

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: Daisuke IMAMURA

GAU:

SERIAL NO: 09/805,948

EXAMINER:

FILED: March 15, 2001

FOR: COMPUTER PROGRAM PRODUCT, RECORDING MEDIUM HAVING SCREEN COMPONENT  
INTERFACE PROGRAM CODE RECORDED THEREIN, AND SCREEN PROGRAM CREATING METHOD

REQUEST FOR PRIORITY

ASSISTANT COMMISSIONER FOR PATENTS  
WASHINGTON, D.C. 20231

SIR:

- ☐ Full benefit of the filing date of U.S. Application Serial Number [US App No], filed [US App Dt], is claimed pursuant to the provisions of **35 U.S.C. §120**.
- ☐ Full benefit of the filing date of U.S. Provisional Application Serial Number , filed , is claimed pursuant to the provisions of **35 U.S.C. §119(e)**.
- ☒ Applicants claim any right to priority from any earlier filed applications to which they may be entitled pursuant to the provisions of **35 U.S.C. §119**, as noted below.

In the matter of the above-identified application for patent, notice is hereby given that the applicants claim as priority:

<u>COUNTRY</u>	<u>APPLICATION NUMBER</u>	<u>MONTH/DAY/YEAR</u>
JAPAN	2000-132694	May 1, 2000
JAPAN	2001-048494	February 23, 2001

Certified copies of the corresponding Convention Application(s)

- ☒ are submitted herewith
- ☐ will be submitted prior to payment of the Final Fee
- ☐ were filed in prior application Serial No. filed
- ☐ were submitted to the International Bureau in PCT Application Number .  
Receipt of the certified copies by the International Bureau in a timely manner under PCT Rule 17.1(a) has been acknowledged as evidenced by the attached PCT/IB/304.
- ☐ (A) Application Serial No.(s) were filed in prior application Serial No. filed ; and  
(B) Application Serial No.(s)
- ☐ are submitted herewith
- ☐ will be submitted prior to payment of the Final Fee

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,  
MAIER & NEUSTADT, P.C.

  
Marvin J. Spivak

Registration No. 24,913

Joseph A. Scafetta, Jr.  
Registration No. 26,803



22850

Tel. (703) 413-3000  
Fax. (703) 413-2220  
(OSMMN 10/98)

07/805, 948

日 本 国 特 許 庁  
PATENT OFFICE  
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

2001年 2月23日

出 願 番 号

Application Number:

特願2001-048494

出 願 人

Applicant (s):

株式会社東芝

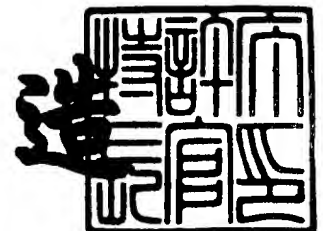


CERTIFIED COPY OF  
PRIORITY DOCUMENT

2001年 3月16日

特許庁長官  
Commissioner,  
Patent Office

及 川 耕 造



【書類名】 特許願

【整理番号】 A000006633

【提出日】 平成13年 2月23日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 3/00

【発明の名称】 画面制御プログラム、動的表示情報取得プログラム、画面表示トランザクションプログラム、画面コンポーネントインタフェースプログラム、画面プログラム作成方法

【請求項の数】 9

【発明者】

【住所又は居所】 東京都府中市東芝町 1 番地 株式会社東芝府中事業所内

【氏名】 今村 大輔

【特許出願人】

【識別番号】 000003078

【氏名又は名称】 株式会社 東芝

【代理人】

【識別番号】 100058479

【弁理士】

【氏名又は名称】 鈴江 武彦

【電話番号】 03-3502-3181

【選任した代理人】

【識別番号】 100084618

【弁理士】

【氏名又は名称】 村松 貞男

【選任した代理人】

【識別番号】 100068814

【弁理士】

【氏名又は名称】 坪井 淳

【選任した代理人】

【識別番号】 100092196

【弁理士】

【氏名又は名称】 橋本 良郎

【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

【選任した代理人】

【識別番号】 100070437

【弁理士】

【氏名又は名称】 河井 将次

【先の出願に基づく優先権主張】

【出願番号】 特願2000-132694

【出願日】 平成12年 5月 1日

【手数料の表示】

【予納台帳番号】 011567

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9705037

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 画面制御プログラム、動的表示情報取得プログラム、画面表示トランザクションプログラム、画面コンポーネントインタフェースプログラム、画面プログラム作成方法

【特許請求の範囲】

【請求項 1】 コンピュータに、

動的に変化する表示内容を得るために予め用意されている動的表示処理を指定する動的表示処理識別情報と、この動的表示処理識別情報で指定されている動的表示処理で利用する表示属性情報とを取得する動的表示情報取得機能、

前記動的表示情報取得機能によって取得された動的表示処理識別情報で指定されている動的表示処理に対してその動的表示処理に対応する前記表示属性情報を提供する属性情報提供機能

を実現させるための画面制御プログラム。

【請求項 2】 請求項 1 記載の画面制御プログラムであって、

前記属性情報提供機能は、前記動的表示情報取得機能によって取得された動的表示処理識別情報で指定されている動的表示処理を組み込むことを特徴とする画面制御プログラム。

【請求項 3】 請求項 1 又は請求項 2 記載の画面制御プログラムであって、

前記動的表示処理は、表示属性情報の内容に基づいて検索を行う処理であることを特徴とする画面制御プログラム。

【請求項 4】 コンピュータに、

動的に変化する表示内容を得るために予め用意されている動的表示処理を指定する動的表示処理識別情報と、この動的表示処理識別情報で指定されている動的表示処理で利用する表示属性情報とを取得する取得機能、

動的表示処理識別情報で指定されている動的表示処理にその動的表示処理で利用する前記表示属性情報を提供する画面制御機能に対して、前記取得機能によって取得された動的表示処理識別情報と表示属性情報とを提供する提供機能を実現させるための画面制御プログラム。

【請求項 5】 請求項 4 記載の動的表示情報取得プログラムであって、

所定のフォーマットで記述されている動的表示処理識別情報と表示属性情報とを前記画面制御機能で取り扱い可能なフォーマットに変換するフォーマット変換機能を、コンピュータに実現させる動的表示情報取得プログラム。

【請求項 6】 コンピュータに、

画面に対する入力内容が正常か否かを判断し、正常であると判断した場合にその入力内容を画面表示トランザクション情報として記憶する記憶機能、

再表示指示を受け付けた場合に、前記記憶機能によって記憶された画面表示トランザクション情報を参照し、任意の時点での正常な入力内容を画面上に再現する再現機能

を実現させるための画面表示トランザクションプログラム。

【請求項 7】 コンピュータに、

画面プログラムを構成するために階層的に組み合わされる画面コンポーネント間で、データを交換するためのインタフェース機能を実現させるための画面コンポーネントインタフェースプログラム。

【請求項 8】 コンピュータに、

画面プログラムを構成するために用意されている画面コンポーネントと、この画面コンポーネントに対して機能とデータとのうちの少なくとも一方を付加するために装着されるカスタムコンポーネントとの間で、データを交換するためのインタフェース機能を実現させるための画面コンポーネントインタフェースプログラム。

【請求項 9】 コンピュータシステムにより画面プログラムを作成する方法において、

画面プログラムを構成するために用意されている画面コンポーネントとその配置位置をユーザに選択させるための画面を、ディスプレイに表示するステップと

ユーザによる画面コンポーネントとその配置位置の選択内容を入力手段により入力するステップと、

ユーザによって選択された画面コンポーネントに対して機能とデータとのうちの少なくとも一方を付加するために装着されるカスタムコンポーネントを、ユー

ザに選択させるための画面をディスプレイに表示するステップと、

ユーザによるカスタムコンポーネントの選択内容を、入力手段により入力するステップと、

ユーザによって選択された画面コンポーネントにカスタムコンポーネントを装着して画面プログラムの構成要素とするステップと  
からなる画面プログラム作成方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、ディスプレイに画面を表示し、ユーザからの対話的な入力を受け付けるための画面プログラムを構成する画面制御プログラム、動的表示情報取得プログラム、画面表示トランザクションプログラム、画面コンポーネントインタフェースプログラム、画面プログラム作成方法に関する。

【0002】

【従来の技術】

ユーザが対話的にデータを入力するための画面プログラムを作成する場合において、プログラム言語により画面プログラムを記述して作成する方法が利用されている。

【0003】

また、他の画面プログラムの作成方法として、開発ツールを用いて画面イメージを作画的に編集しながら画面に関するプログラムソースを生成する方法がある。

【0004】

さらに、ファイルなどに画面のレイアウト情報を定義しておき、実行時にプログラムがこの画面のレイアウト情報を読み込んで画面を表示する方法も利用されている。

【0005】

【発明が解決しようとする課題】

上記の3種類の画面プログラム作成方法のうち、プログラム言語により画面プ

プログラムを記述する方法を利用した場合には、プログラムの自由度は高くなるが記述量が多くなる。また、プログラムを変更する場合にプログラムの編集及びコンパイル等の作業が必要である。したがって、開発効率の向上が困難である。

【 0 0 0 6 】

また、画面プログラムの開発ツールを利用する場合には、画面のレイアウト作成は容易になる。しかしながら、画面表示のために必要とされる処理については、プログラム言語で記述する必要がある。したがって、この方法を利用しても画面プログラムの作成が困難な場合がある。また、この方法ではレイアウトの変更が必要な場合にコンパイルする必要がある。したがって、画面プログラムの作成・変更に手間がかかる。

【 0 0 0 7 】

画面のレイアウト情報を予め定義しておいて実行時に読み込む場合には、画面のレイアウトを変更する場合であってもプログラムソースの編集及びコンパイルが不要である。したがって、この方法を利用すれば、プログラムの開発効率はよくなる。しかしながら、この方法では、プログラム外部に定義できるレイアウト情報の情報量に限りがあり、自由度と拡張性が高くない。例えば、この方法を利用すると、画面を構成する部品の大きさ、位置、色の種類、画面上に配置される文字の指定などについては、レイアウト情報として定義可能である。その一方、この方法を利用した場合には、画面表示のために必要とされる処理についてプログラム内で定義する必要がある、レイアウト情報などのようにプログラム外部で定義できない。したがって、画面表示のために必要とされる処理についてはプログラムで記述する必要がある。また、画面表示のために必要とされる処理を変更する場合には、作成したプログラムをコンパイルする必要がある。

【 0 0 0 8 】

このように、上記 3 種類の画面プログラム作成方法を利用する場合には、画面表示のために必要とされる処理をプログラム言語により記述する必要がある。また、処理の変更時にコンパイルする必要がある。このため、開発労力がかかるといふ問題がある。

【 0 0 0 9 】



さらに、各種のアプリケーション・プログラム（以下、アプリケーションという）で問題になるのは、障害発生時のアプリケーションの状態管理である。業務処理の障害に応じてデータベースの内容を復帰させることは、通常行われている。その一方、業務処理の障害が発生した場合に、画面の入力状態についても同様に復帰させることが望まれる。アプリケーションが障害から復帰した場合に、障害発生前において入力済みの内容を、ユーザが画面に対して再入力するのは効率的ではない。

#### 【0010】

しかしながら、画面の表示状態を適切な状態に復帰させるプログラム開発には多くの労力がかかる。

#### 【0011】

加えて、従来において、アプリケーションの画面プログラムにおいて実行する処理は、データのキー入力受付処理、一覧に基づく選択受付処理、プログラム（業務ロジック）の実行依頼処理などに定形化することができる。

#### 【0012】

ここで、データのキー入力受付処理とは、テキストフィールドを表示し、ユーザのテキスト入力を受け付ける処理である。データのキー入力受付処理は、入力されたテキストのチェックとチェック結果に応じた動作を実行する。

#### 【0013】

一覧に基づく選択受付処理とは、選択肢となる一覧をデータベースやファイルから取得して表示し、ユーザの選択を受け付ける処理である。

#### 【0014】

プログラムの実行依頼処理とは、実行ボタンの押下などにより、入力されたデータを収集し、その収集したデータをプログラムに渡してそのプログラムの実行を依頼する処理、または収集したデータをサーバプログラムに送信する処理である。

#### 【0015】

従来において、データのキー入力受付処理を実行する画面プログラムを作成する場合には、開発者は、ボタン、テキストフィールドなどの画面コンポーネント

の配置を定めた後、部品毎にキー入力イベントにより入力チェックを行うプログラムを記述する必要がある。

【0016】

また、開発者は、一覧に基づく選択受付処理を実行する画面プログラムを作成する場合には、一覧データを取得し、かつ画面コンポーネントに対して取得したデータを設定するためのプログラムを記述する必要がある。

【0017】

また、プログラムの実行を依頼する画面プログラムを作成する場合には、開発者は、ボタンの押下イベントにより画面に配置された各画面コンポーネントのデータを取得し、プログラム又はサーバに渡す形式に変換し、プログラム又はサーバを呼び出すためのプログラムを記述する必要がある。

【0018】

しかしながら、画面プログラムを作成する度にこのような作業を行うのは労力がかかる。

【0019】

さらに、複雑な画面プログラムを作成する場合には、1画面を複数の開発者で分担して作成する必要がある。しかしながら、従来は、画面を複数の部分に分割し、それぞれの部分について画面プログラムを作成することが困難であった。

【0020】

本発明は、以上のような実情に鑑みてなされたもので、拡張性の高い画面プログラムを効率的かつ容易に開発し、画面プログラムの耐障害性を向上させるための画面制御プログラム、動的表示情報取得プログラム、画面表示トランザクションプログラム、画面コンポーネントインタフェースプログラム、画面プログラム作成方法を提供することを目的とする。

【0021】

【課題を解決するための手段】

本発明を実現するにあたって講じた具体的手段について以下に説明する。

【0022】

第1の発明は、コンピュータに、動的に変化する表示内容を得るために予め用

意されている動的表示処理を指定する動的表示処理識別情報と、この動的表示処理識別情報で指定されている動的表示処理で利用する表示属性情報とを取得する動的表示情報取得機能、動的表示情報取得機能によって取得された動的表示処理識別情報で指定されている動的表示処理に対してその動的表示処理に対応する表示属性情報を提供する属性情報提供機能、を実現させるための画面制御プログラムである。

【 0 0 2 3 】

この第 1 の発明の骨子は、動的に変化する表示内容を得るために動的表示処理が予め用意されており、この動的表示処理の指定及びこの動的表示処理で利用する属性の指定によって画面プログラムを構成可能な点にある。

【 0 0 2 4 】

この第 1 の発明における画面制御プログラムを利用する場合においては、開発者は、動的表示処理と表示属性情報とを指定すればよい。これにより、開発者は、表示する時点で動的に変化する内容を画面上に表示させる処理をプログラム言語により記述する必要がなく、容易に画面プログラムを作成できる。また、開発者は、画面プログラムの作成時あるいは変更時に、作成中の画面プログラムをコンパイルする必要もない。

【 0 0 2 5 】

したがって、開発者は、拡張性の高い画面プログラムを効率的に開発できる。

【 0 0 2 6 】

第 2 の発明は、第 1 の発明と同様の画面制御プログラムである。この第 2 の発明における属性情報提供機能は、動的表示情報取得機能によって取得された動的表示処理識別情報で指定されている動的表示処理を組み込む。

【 0 0 2 7 】

この第 2 の発明においては、予め用意されている動的表示処理のうち、開発者によって指定された動的表示処理が属性情報提供手段に対してプラグインされ、表示属性情報に基づく利用が可能となる。

【 0 0 2 8 】

これにより、開発者は、画面プログラム作成時あるいは変更時にプログラムを

記述する労力を削減できる。

【 0 0 2 9 】

すなわち、開発者は、単に動的表示処理と表示属性情報を指定するのみで、画面プログラムに組み込むべき処理を利用・変更できる。

【 0 0 3 0 】

第 3 の発明は、第 1 又は第 2 の発明と同様の画面制御プログラムである。この第 3 の発明における動的表示処理は、表示属性情報の内容に基づいて検索を行う処理であるという特徴を持つ。

【 0 0 3 1 】

これにより、あるキーにしたがって検索されたデータを画面に表示する画面プログラムを作成する場合に、開発者は、検索を行う動的表示処理を指定し、そのキーを表示属性情報として指定するのみでよい。したがって、上記と同様の効果を得ることができる。

【 0 0 3 2 】

なお、動的表示処理の他の例としては、入力内容をチェックしてエラーの場合にその旨を返す入力チェック処理、新規の画面をポップアップするポップアップ処理などがある。

【 0 0 3 3 】

第 4 の発明は、コンピュータに、動的に変化する表示内容を得るために予め用意されている動的表示処理を指定する動的表示処理識別情報と、この動的表示処理識別情報で指定されている動的表示処理で利用する表示属性情報とを取得する取得機能、動的表示処理識別情報で指定されている動的表示処理にその動的表示処理で利用する表示属性情報を提供する画面制御機能に対して、取得機能によって取得された動的表示処理識別情報と表示属性情報とを提供する提供機能、を実現させるための画面制御プログラムである。

【 0 0 3 4 】

この第 4 の発明における動的表示情報取得プログラムは、先の第 1 の発明における動的表示情報取得機能としてコンピュータを機能させるためのプログラムである。

## 【 0 0 3 5 】

第 5 の発明は、第 4 の発明と同様の動的表示情報取得プログラムである。この第 5 の発明における動的表示情報取得プログラムは、所定のフォーマットで記述されている動的表示処理識別情報と表示属性情報とを画面制御機能で取り扱い可能なフォーマットに変換するフォーマット変換機能をコンピュータに実現させる。

## 【 0 0 3 6 】

これにより、例えばXML (Extensible Markup Language) ファイル、HTML (HyperText Markup Language) ファイル、CSVファイル等のような所定の形式で記述されたファイルとして動的表示処理識別情報と表示属性情報とを受け付け、その内容を後段の画面制御機能で取り扱い可能に変換し、変換後のファイルを画面制御機能に提供できる。

## 【 0 0 3 7 】

第 6 の発明は、コンピュータに、画面に対する入力内容が正常か否かを判断し、正常であると判断した場合にその入力内容を画面表示トランザクション情報として記憶する記憶機能、再表示指示を受け付けた場合に、記憶機能によって記憶された画面表示トランザクション情報を参照し、任意の時点での正常な入力内容を画面上に再現する再現機能、を実現させるための画面表示トランザクションプログラムである。

## 【 0 0 3 8 】

この第 6 の発明における画面表示トランザクションプログラムは、上記の動的表示処理の一つとして利用可能である。この画面表示トランザクションプログラムを利用すると、画面への入力作業中に障害が発生し、その後障害から復帰した場合に、障害発生前の任意の時点での正常な入力内容を再表示することができる。

## 【 0 0 3 9 】

したがって、ユーザの操作を効率化でき、画面プログラムの耐障害性を向上させることができる。

## 【 0 0 4 0 】

第7の発明は、コンピュータに、画面プログラムを構成するために階層的に組み合わされる画面コンポーネント間で、データを交換するためのインタフェース機能を実現させるための画面コンポーネントインタフェースプログラムである。

## 【0041】

この第7の発明を利用することにより、管理型の画面コンポーネントと非管理型の画面コンポーネントとの間のデータ交換を容易に行うことができ、画面プログラムの作成を容易化できる。

## 【0042】

第8の発明は、コンピュータに、画面プログラムを構成するために用意されている画面コンポーネントと、この画面コンポーネントに対して機能とデータとのうちの少なくとも一方を付加するために装着されるカスタムコンポーネントとの間で、データを交換するためのインタフェース機能を実現させるための画面コンポーネントインタフェースプログラムである。

## 【0043】

この第8の発明を利用することにより、画面コンポーネントとこの画面コンポーネントに装着されるカスタムコンポーネントとの間のデータ交換を容易に行うことができる。また、カスタムコンポーネントの作成が容易になる。したがって、画面プログラムの作成を容易化できる。

## 【0044】

なお、上記のようなプログラム、及びこのプログラムを記録した記録媒体を用いることによって、上述した機能を有していない計算機システム、サーバやクライアント等の計算機に対して、簡単に上述した機能を付加することができる。

## 【0045】

第9の発明は、コンピュータシステムにより画面プログラムを作成する方法に関する。この第9の発明の画面プログラム作成方法は、画面プログラムを構成するために用意されている画面コンポーネントとその配置位置をユーザに選択させるための画面をディスプレイに表示するステップと、ユーザによる画面コンポーネントとその配置位置の選択内容を入力手段により入力するステップと、ユーザによって選択された画面コンポーネントに対して機能とデータとのうちの少なく

とも一方を付加するために装着されるカスタムコンポーネントを、ユーザに選択させるための画面をディスプレイに表示するステップと、ユーザによるカスタムコンポーネントの選択内容を入力手段により入力するステップと、ユーザによって選択された画面コンポーネントにカスタムコンポーネントを装着して画面プログラムの構成要素とするステップとからなる。

【0046】

この方法を利用することで、開発者の画面プログラムの作成が容易になる。

【0047】

【発明の実施の形態】

以下、図面を参照しながら本発明の実施の形態について説明する。なお、以下に示す各図において、同一の部分については同一の符号を付してその説明を省略し、ここでは異なる部分についてのみ詳しく説明する。

【0048】

(第1の実施の形態)

本実施の形態においては、レイアウト情報に加えて、画面表示のために必要とされる処理を指定する動的表示情報を取得し、アプリケーション（業務アプリケーション）の画面を表示する画面制御プログラムについて説明する。

【0049】

図1は、本実施の形態に係る画面制御プログラムによって計算機上で実現される機能を示すブロック図である。

【0050】

本実施の形態に係る画面制御プログラム1は、主にレイアウト情報取得機能2、動的表示情報取得機能3、制御機能4を計算機や計算機システム上で実現する。また、画面制御プログラム1は、検索処理インタフェース5a、入力チェック処理インタフェース5b、ポップアップ処理インタフェース5cを持つ。さらに、画面制御プログラム1は、データ入出力機能6、画面表示トランザクション機能7を計算機や計算機システム上で実現する。

【0051】

画面制御プログラム1は、例えば検索処理や、入力チェック処理、ポップアッ

ブ処理などのように、画面に表示される時点で動的に変化する表示内容を得るために予め作成されている動的表示処理（以下、「コンポーネント」という）8 a ～ 8 c を利用する。コンポーネント 8 a ～ 8 c は既にコンパイルされており、そのままの状態で作成可能なプログラムである。

#### 【 0 0 5 2 】

開発者（画面表示プログラム作成者）は、アプリケーション 1 1 の画面についてのレイアウト情報 9 と動的表示情報 1 0 とを入力する。

#### 【 0 0 5 3 】

レイアウト情報 9 には、先に述べたように、例えば画面を構成するための部品の配置位置、大きさ、色などのような、画面を構成する場合に固定的に決定される静的情報が主に記述されている。

#### 【 0 0 5 4 】

動的表示情報 1 0 には、ユーザが利用したいコンポーネントを指定した動的表示処理識別情報や、このコンポーネントの処理内容を決定するために必要な表示属性情報が記述されている。ここでは、例として検索処理、入力チェック処理、ポップアップ処理を指定する旨の動的表示処理識別情報が、動的表示情報 1 0 に記述されている。また、動的表示情報 1 0 には、検索処理においてテーブル  $T_1$  の内容が検索対象となる旨の表示属性情報が記述されている。

#### 【 0 0 5 5 】

レイアウト情報取得機能 2 は、レイアウト情報 9 に定義されている内容を解析し、解析結果を制御機能 4 に伝える。

#### 【 0 0 5 6 】

動的表示情報取得機能 3 は、動的表示情報 1 0 に定義された内容を解析し、解析結果を制御機能 4 に伝える。この動的表示情報取得機能 3 は、主に取得機能 3 a と、変換機能 3 b と、提供機能 3 c とから構成されている。

#### 【 0 0 5 7 】

所定のフォーマットで記述されている動的表示情報 1 0 が取得機能 3 a によって読み込まれると、その記述内容は変換機能 3 b によって制御機能 4 で取り扱い可能なフォーマットに変換され、提供機能 3 c によって制御機能 4 に提供される



## 【 0 0 5 8 】

制御機能 4 の備える属性情報提供機能 4 a は、動的表示情報取得機能 3 から提供された動的表示処理識別情報によって指定されているコンポーネントを、プラグインするための制御を行う。また、属性情報提供機能 4 a は、プラグインされたコンポーネントに対して該当する表示属性情報を提供し、結果を取得する。

## 【 0 0 5 9 】

加えて、制御機能 4 は、フィールド（画面上で入出力を行うための部品、例えばテキストフィールド）の表示や動作の制御を行う。

## 【 0 0 6 0 】

検索処理インタフェース 5 a は、検索処理用のコンポーネント 8 a をプラグインし、動的表示情報 1 0 で定義された検索用の表示属性情報に基づいてファイルやデータベースのテーブルを検索し、検索結果を制御機能 4 に返却する。

## 【 0 0 6 1 】

入力チェック処理インタフェース 5 b は、入力チェック処理用のコンポーネント 8 b をプラグインし、レイアウト情報 9 もしくは動的表示情報 1 0 に定義された入力チェック用の表示属性情報に基づいてフィールドに入力された値をチェックし、そのチェック結果を制御機能 4 に返却する。

## 【 0 0 6 2 】

ポップアップ処理インタフェース 5 c は、ポップアップ処理用のコンポーネント 8 c をプラグインし、動的表示情報 1 0 で定義されたポップアップ処理用の表示属性情報に基づいてポップアップウィンドウを表示し、ポップアップウィンドウ上で決定されたデータを制御機能 4 に反映する。

## 【 0 0 6 3 】

データ入出力機能 6 は、画面上のフィールドに対して入力されたデータを取得し、またアプリケーション 1 1 から提供されたデータを画面上のフィールドにセットする。

## 【 0 0 6 4 】

画面表示トランザクション処理機能 7 は、入出力の状態が正常な場合にその入

出力状態を示すオブジェクトを作成し、保存する。また、画面に対するトランザクションの開始を指示された場合に、保存されているオブジェクトを用いて画面表示の復帰を行う。

## 【 0 0 6 5 】

以下に、上記構成により実現される動作を具体的に説明する。

## 【 0 0 6 6 】

図 2 は、レイアウト情報 9 の具体例を示す図である。このレイアウト情報 9 の例では、XML を用いて記述されている場合を示しているが、その他にも CSV 形式、特定の文字をデリミタとする形式、固定長ファイル形式が利用できる。

## 【 0 0 6 7 】

図 3 は、本実施の形態に係る画面制御プログラム 1 がレイアウト情報 9 を読み込んだ場合の処理手順を例示するシーケンス図である。

## 【 0 0 6 8 】

まず、アプリケーション 1 1 が、レイアウト情報取得機能 2 に対してレイアウト情報 9 をセットする ( 3 0 1 ) 。レイアウト情報取得機能 2 は、先の図 2 に示すレイアウト情報 9 の 2 行目～ 2 1 行目に示す <field> ～ </field> を 1 フィールド分の定義として展開する ( 3 0 2 ) 。XML 文書の展開には XML パーザといった既存の技術を用いることができる。

## 【 0 0 6 9 】

展開されたフィールド毎のレイアウト情報は、レイアウト情報取得機能 2 から制御機能 4 に対してセットされる ( 3 0 3 ) 。

## 【 0 0 7 0 】

制御機能 4 は、セットされたレイアウト情報にしたがってフィールドを作成する ( 3 0 4 ) 。以下、処理 3 0 2 ～ 3 0 4 が <field> タグの数だけ繰り返される ( 3 0 5 ) 。

## 【 0 0 7 1 】

図 4 は、<field> タグの数だけ繰り返される各フィールドの作成処理を例示するフローチャートである。

## 【 0 0 7 2 】

制御機能 4 は、先の図 2 に示すレイアウト情報 9 の 2 行目の <field> タグ内に指定されている ID を関連付けたフィールドオブジェクトを作成する (4 0 1)。

#### 【0 0 7 3】

次に、先の図 2 に示すレイアウト情報 9 の 3 行目～1 1 行目に定義されている位置属性などのフィールドの表示に関する情報を、作成されたフィールドオブジェクトに対してセットする (4 0 2)。

#### 【0 0 7 4】

例えば、制御機能 4 は、先の図 2 に示すレイアウト情報 9 の 7 行目の <label> タグに囲まれた属性により、フィールドのラベルをフィールドオブジェクトにセットする。

#### 【0 0 7 5】

次に、制御機能 4 は、レイアウト情報 9 に入力チェック属性が定義されているか否かを判断する (4 0 3)。先の図 2 に示すレイアウト情報 9 については、1 3 行目の <validation> タグの有無により判断可能である。

#### 【0 0 7 6】

ここで、入力チェック属性が指定されている場合には、先の図 2 に示すレイアウト情報 9 の 1 5 行目の <component> タグで指定された入力チェック用のコンポーネントを作成するか、あるいはコンポーネントのインスタンスを取得する (4 0 4)。J a v a 言語の場合、クラス名からオブジェクトの作成が可能であり、この処理 4 0 4 に用いることができる。

#### 【0 0 7 7】

次に、制御機能 4 は、入力チェック用コンポーネントをフィールドに登録する (4 0 5)。

#### 【0 0 7 8】

次に、制御機能 4 は、先の図 2 に示すレイアウト情報 9 の 1 4 行目に示す入力チェックのトリガーとなるキーと、1 6 行目～1 9 行目に示すエラー時の動作とをフィールドに登録する (4 0 6)。

#### 【0 0 7 9】

そして、制御機能 4 は、フィールドを先の図 2 に示すレイアウト情報 9 の 1 2 行目に指定するように初期状態にし、処理を完了する（4 0 7）。この例では、初期状態は「UNEDITABLE」と指定される。

【0 0 8 0】

図 5 は、上記のようなレイアウト情報 9 に基づく制御機能 4 の処理によって構成される画面を示す図である。

【0 0 8 1】

レイアウト情報 9 に基づいて作成された画面 1 2 は、ラベル 1 2 a ～ 1 2 c や初期状態のフィールド 1 2 d ～ 1 2 f などのような画面要素を表示している。

【0 0 8 2】

例えば、先の図 2 に示すレイアウト情報 9 の 7 行目の <label> タグに囲まれた属性は、フィールド 1 2 d のラベル 1 2 a に相当する。このため、ラベル 1 2 a には「数量」と表示される。

【0 0 8 3】

また、先の図 2 に示すレイアウト情報 9 の 1 2 行目において、初期状態は「UNEDITABLE」と指定する旨が記載されているため、フィールド 1 2 d の初期状態は編集不可能状態となっている。

【0 0 8 4】

図 6 は、画面制御プログラム 1 がレイアウト情報 9 を読み込んだ場合の処理手順の変形例を示すシーケンス図である。

【0 0 8 5】

この変形例において、アプリケーション 1 1 は、レイアウト情報取得機能 2 に対してレイアウト情報 9 をセットする（6 0 1）。

【0 0 8 6】

レイアウト情報取得機能 2 は <field> ～ </field> までは 1 フィールド分の定義として展開する（6 0 2）。

【0 0 8 7】

次に、レイアウト情報取得機能 2 が 1 フィールド分のレイアウト情報を格納したレイアウト情報オブジェクト 9 4 を作成し（6 0 3、6 0 4）、このレイアウ

ト情報オブジェクト 9 4 を制御機能 4 にセットする (6 0 5)。

【0 0 8 8】

制御機能 4 は、フィールドを作成し、属性をセットする (6 0 6)。

【0 0 8 9】

その後、処理 6 0 2 から処理 6 0 6 までの動作がフィールドの数だけ繰り返される (6 0 7)。

【0 0 9 0】

図 7 は、レイアウト情報 9 において定義された入力チェック属性にしたがって画面制御プログラム 1 が実行する入力チェック処理の処理手順を例示するシーケンス図である。

【0 0 9 1】

ユーザは、フィールドに値を入力した後、入力チェックのトリガーとなる操作を行う (7 0 1)。先の図 2 に示すレイアウト情報 9 の 1 4 行目に <trigger> タグで囲まれた属性として「VK\_ENTER」が指定されている。この「VK\_ENTER」は、エンターキーの入力が入力チェックのトリガーになることを示している。

【0 0 9 2】

このトリガーキーの入力によって、制御機能 4 は、フィールドに入力された値を入力チェック処理インタフェース 5 b に対して渡し、入力のチェックを指示する (7 0 2)。

【0 0 9 3】

入力チェック処理インタフェース 5 b は、値の入力されたフィールドに登録された入力チェック処理用のコンポーネント 8 b に対して入力チェックを指示する (7 0 3)。

【0 0 9 4】

入力チェック処理用のコンポーネント 8 b は、渡された値を自分の実装により判断し、エラーか否かを入力チェック処理インタフェース 5 b に返却する (7 0 4)。

【0 0 9 5】

入力チェック処理インタフェース 5 b は、得られた結果を制御機能 4 に返却す

る (705)。

【0096】

制御機能4は、チェック結果がエラーだった場合に先の図2に示す16行目～19行目に定義された内容にしたがってフィールドの属性変更を行う。

【0097】

このような動作は、共通的に必要とされる性質であるためフィールドを構成するプログラムに予め用意されている。レイアウト情報9が記述されているファイルでは、作り込まれているプログラムの中から選択的に利用する処理を指定している。この例では、〈type〉タグに「CHANGE\_BACKGROUND\_COLOR」と定義されており、また〈param〉タグで「background」が#FF0000と指定されている。

【0098】

したがって、図8に例示するように、入力チェックの結果がエラーだった場合には、そのフィールド12dの背景色が赤になる。

【0099】

続いて、本実施の形態において特徴的な情報である動的表示情報10について説明する。なお、上記のレイアウト情報9において説明した入力チェック属性は、この動的表示情報10で指定されてもよい。

【0100】

図9は、動的表示情報10の具体例を示す図である。

【0101】

この動的表示情報10の例では、レイアウト情報9と同様にXMLを用いて記述された場合を示しているが、他にもCSV形式、特定の文字をデリミタ (Delimiter) とする形式、固定長ファイル形式で記述してもよい。

【0102】

図10は、本実施の形態に係る画面制御プログラム1が動的表示情報10を読み込んだ場合の処理手順を例示するシーケンス図である。

【0103】

まず、アプリケーション11が、動的表示情報取得機能3に対して動的表示情報10をセットする(1001)。動的表示情報取得機能3は、先の図9に示す

動的表示情報 1 0 の 2 行目～1 3 行目に示す <field> ～ </field> を 1 フィールド分の定義として展開する（1 0 0 2）。XML 文書の展開にはレイアウト情報 9 の場合と同様に XML パーザといった既存の技術を用いることができる。

#### 【0 1 0 4】

展開されたフィールド毎の動的表示情報は、動作表示情報取得機能 3 から制御機能 4 に対してセットされる（1 0 0 3）。

#### 【0 1 0 5】

制御機能 4 は、セットされた動的表示情報にしたがってフィールドの表示属性をセットする（1 0 0 4）。以下、処理 1 0 0 2 ～ 1 0 0 4 の処理が <field> タグの数だけ繰り返される（1 0 0 5）。この結果、画面上のそれぞれのフィールドについての表示属性が変更され、かつ動作に必要な情報が各フィールドにセットされる。

#### 【0 1 0 6】

図 1 1 は、上記のような動的表示情報 1 0 に基づく制御機能 4 の処理によって構成される画面を示す図である。

#### 【0 1 0 7】

この画面 1 2 においては、先の図 9 に示す動的表示情報 1 0 における 3 行目の指定にしたがって、「area」という ID のフィールド 1 2 e が <state> タグで囲まれた「COMBO」という表示属性となる。この「COMBO」という表示属性は、一覧に基づく選択を可能とし、かつフィールドに対する直接の文字入力を可能とするコンポーネントを示す動的表示処理識別情報である。「COMBO」という表示属性を指定することで、画面要素 1 4 a に示すような動的な表示形態が採用される。

#### 【0 1 0 8】

また、先の図 9 に示す動的表示情報 1 0 における 1 5 行目の指定にしたがって、「product」という ID を持つフィールド 1 2 f が「POPUP」という表示属性となる。この「POPUP」はフィールドにポップアップウィンドウを表示する時に押下するためのボタンを付属するコンポーネントを示す動的表示処理識別情報である。「POPUP」という表示属性を指定することで、画面要素 1 4 b に示すような

動的な表示形態が採用される。このフィールドにセットされる表示属性毎の処理については後述する。

## 【 0 1 0 9 】

図 1 2 は、画面制御プログラム 1 が動的表示情報 1 0 を読み込んだ場合の処理手順の変形例を示すシーケンス図である。

## 【 0 1 1 0 】

この変形例において、アプリケーション 1 1 は、動的表示情報取得機能 3 に対して動的表示情報 1 0 をセットし ( 1 2 0 1 ) 、 <field> ~ </field> までも 1 フィールド分の定義として展開する ( 1 2 0 2 ) 。

## 【 0 1 1 1 】

次に、動的表示情報取得機能 3 は、1 フィールド分の動的表示情報の格納された動的表示情報オブジェクト 9 5 を作成し ( 1 2 0 3 、 1 2 0 4 ) 、この動的表示情報オブジェクト 9 5 を制御機能 4 にセットする ( 1 2 0 5 ) 。

## 【 0 1 1 2 】

制御機能 4 は、フィールドに属性をセットする ( 1 2 0 6 ) 。

## 【 0 1 1 3 】

そして、処理 1 2 0 2 から 1 2 0 6 と同様の動作がフィールドの数だけ繰り返される ( 1 2 0 7 ) 。

## 【 0 1 1 4 】

図 1 3 は、フィールドにセットされた表示属性が検索だった場合に画面制御プログラム 1 が実行する処理の手順を例示するシーケンス図である。

## 【 0 1 1 5 】

また、図 1 4 は、フィールドにセットされた表示属性が検索だった場合の画面制御プログラム 1 の処理を例示するフローチャートである。

## 【 0 1 1 6 】

制御機能 4 に動的表示情報 1 0 がセットされると、制御機能 4 はそれぞれのフィールド毎にフィールドの表示属性を変更する ( 1 4 0 1 ) 。

## 【 0 1 1 7 】

制御機能 4 は、フィールドの表示属性が「COMBO」、「PULLDOWN」といった一



覧選択を行うものであるか否か（検索についての動的表示処理識別情報か否か）を判断する（1402）。

【0118】

制御機能4は、フィールドの表示属性が一覧選択を行うものである場合に、一覧に格納するデータを検索するために必要な属性（表示属性情報）が動的表示情報にあるか否かを判断する（1403）。

【0119】

検索に必要な属性がある場合には、制御機能4は、検索処理インタフェース5aにテーブル名、検索するカラム名、検索条件といった検索のための属性を渡して検索の指示を出す（1301）。

【0120】

検索処理インタフェース5aは、プラグインされた検索処理用のコンポーネント8aを用いて検索処理を行う（1302）。プラグインされた検索処理用のコンポーネント8aは、渡された検索のための属性を利用し、自らの実装に基づいてデータベースやファイルのテーブルを検索する（1303、1404）。

【0121】

検索結果は、検索処理用のコンポーネント8aから検索処理インタフェース5aを経由し（1304）、制御機能4に渡され（1305）、フィールドの一覧データに格納される（1306、1405）。

【0122】

上記の処理の結果、図15に示すように、検索処理用のコンポーネント8aが動的な表示内容を取得する。画面12のフィールド12eはユーザの操作に基づいて検索された一覧15aを表示する。

【0123】

図16は、フィールドにセットされた表示属性がポップアップだった場合に画面制御プログラム1が実行する処理の手順を例示するシーケンス図である。

【0124】

制御機能4に動的表示情報がセットされると、制御機能4は、検索処理の場合と同様にフィールドの表示属性を変更する。また、制御機能4は、フィールドの

表示属性がポップアップを示すものか否か（ポップアップについての動的表示処理識別情報か否か）を判断する。例えば、先の図 9 に示す動的表示情報 1 0 における 1 5 行目に指定する「POPUP」がポップアップを示す表示属性である。

#### 【 0 1 2 5 】

表示属性がポップアップの場合には、制御機能 4 は、ポップアップウィンドウに表示するコンポーネント名、ポップアップウィンドウのタイトルバー部分に表示するタイトル、大きさ、ポップアップウィンドウに付属するボタンのラベル名といった属性（表示属性情報）をポップアップ処理インタフェース 5 c に対して渡し、ポップアップ処理 8 c の登録を指示する（1 6 0 1）。

#### 【 0 1 2 6 】

ポップアップ処理インタフェース 5 c に渡す属性は、先の図 9 に示す動的 1 8 行目～2 2 行目に記述するように定義可能である。

#### 【 0 1 2 7 】

ポップアップ処理インタフェース 5 c は、プラグインされたポップアップ用のコンポーネント 8 c を作成し、インスタンスを取得する（1 6 0 2、1 6 0 3）。また、ポップアップ処理インタフェース 5 c は、取得したポップアップ用のコンポーネント 8 c の表示処理を、フィールドへの操作の結果起動される処理として制御機能 4 に登録する（1 6 0 4）。これは J a v a 言語を例にとると、ボタン押下のイベントにより起動するモジュールをボタンに登録するという処理で実現可能である。

#### 【 0 1 2 8 】

図 1 7 は、ユーザによるフィールドへの操作によって起動されるポップアップ処理の手順を例示するシーケンス図である。

#### 【 0 1 2 9 】

ユーザは、制御機能 4 により表示されているフィールドに対し、ポップアップ表示を命ずる操作を行う（1 7 0 1）。

#### 【 0 1 3 0 】

制御機能 4 は、ポップアップ処理インタフェース 5 c に対し、ポップアップウィンドウの表示を指示する（1 7 0 2）。

## 【 0 1 3 1 】

ポップアップ処理インタフェース 5 c は、プラグインされたポップアップ処理用のコンポーネント 8 c をウィンドウ上に表示する ( 1 7 0 3 ) 。

## 【 0 1 3 2 】

この状態での画面の具体例を図 1 8 に示す。この図 1 8 における画面 1 2 の要素 1 6 a がプラグインされたポップアップ処理用のコンポーネント 8 c である。

## 【 0 1 3 3 】

ここで、ユーザからの入力操作を待ち、ユーザはポップアップウィンドウに対してデータを入力する ( 1 7 0 4 ) 。図 1 8 の例では、画面要素 1 6 b における選択操作がこの入力操作に該当する。

## 【 0 1 3 4 】

ユーザによってポップアップウィンドウに付属している決定ボタンの押下などの確定操作が行われると ( 1 7 0 5 ) 、ポップアップ処理インタフェース 5 c はポップアップウィンドウを閉じる ( 1 7 0 6 ) 。

## 【 0 1 3 5 】

ポップアップウィンドウに付属している決定ボタンとは、図 1 8 の画面要素 1 6 c のように、ポップアップウィンドウ自体への操作のために予め用意されているボタンである。

## 【 0 1 3 6 】

そして、ポップアップ処理インタフェース 5 c は、プラグインされたポップアップ処理用のコンポーネント 8 c に対し、ユーザが入力したデータを取得するための指示を発行する ( 1 7 0 7 ) 。

## 【 0 1 3 7 】

その結果、ポップアップ処理用のコンポーネント 8 c は、入力されたデータの ID とデータの値をポップアップ処理インタフェース 5 c に返却する ( 1 7 0 8 ) 。ここでデータの ID とは、ポップアップ処理インタフェース 5 c とポップアップ処理用のコンポーネント 8 c との間の規約により必要とされるもので、制御機能 4 の任意のフィールドにつけられた ID と同値である。

## 【 0 1 3 8 】

ポップアップ処理インタフェース 5 c は、得られたデータを表示するフィールドをデータの ID をキーにして検索し、制御機能 4 の該当するフィールドにセットする (1 7 0 9)。

【 0 1 3 9 】

この状態での画面の具体例を図 1 9 に示す。この図 1 9 における画面 1 2 のフィールド 1 2 c には、ユーザがポップアップウィンドウで選択したデータがセットされ、表示される。

【 0 1 4 0 】

図 2 0 は、画面表示トランザクション機能 7 を利用して画面表示を復帰させる処理手順を例示するシーケンス図である。

【 0 1 4 1 】

アプリケーション 1 1 は、画面表示トランザクション機能 7 に対して画面表示トランザクションの開始を指示する (2 0 0 1)。

【 0 1 4 2 】

画面表示トランザクション機能 7 は、制御機能 4 に対し、現在の画面状態の取得を指示する (2 0 0 2)。

【 0 1 4 3 】

制御機能 4 は、現在の画面状態を示す情報として、セットされているレイアウト情報、セットされている動的表示情報、各フィールドの活性データを格納した画面表示トランザクション情報オブジェクト 9 6 を作成する (2 0 0 3、2 0 0 4)。ここで、活性データとは、ユーザがフィールドに入力したデータやアプリケーションの実行によってフィールドに表示されたデータである。

【 0 1 4 4 】

制御機能 4 は、画面表示トランザクション情報オブジェクト 9 6 を画面表示トランザクション機能 7 に返却する (2 0 0 5)。

【 0 1 4 5 】

画面表示トランザクション機能 7 はアプリケーション 1 1 に画面表示トランザクション情報オブジェクト 9 6 を返却する (2 0 0 6)。

【 0 1 4 6 】

この時点での画面表示トランザクション情報オブジェクト 9 6 は、表 1 に示すようにレイアウト情報、動的表示情報、活性データを管理している。

【0 1 4 7】

【表 1】

表 1

レイアウト情報	[レイアウト情報オブジェクト]
動的表示情報	[動的表示情報オブジェクト]
活性データ	[活性データの一覧オブジェクト]

【0 1 4 8】

なお、画面表示トランザクション情報オブジェクト 9 6 自体の返却に代えて、画面表示トランザクション情報オブジェクト 9 6 を画面表示トランザクション機能 7 あるいは制御機能 4 によって特定の I D と対応づけて管理し、アプリケーション 1 1 に I D のみを返却するとしてもよい。

【0 1 4 9】

その後、アプリケーション 1 1 は、業務処理を続け（2 0 0 7）、業務処理が失敗し画面状態の復帰が必要になった場合に、画面表示トランザクション機能 7 に対しトランザクションの復帰を指示する（2 0 0 8）。この時、アプリケーション 1 1 は、画面表示トランザクション情報オブジェクト 9 6、もしくは画面表示トランザクション情報オブジェクト 9 6 を示す I D を画面表示トランザクション機能 7 に渡し、復帰を指示する。

【0 1 5 0】

画面表示トランザクション機能 7 は、渡された画面表示トランザクション情報オブジェクト 9 6、もしくは画面表示トランザクション情報オブジェクト 9 6 を示す I D を制御機能 4 にセットして状態の復帰を指示する（2 0 0 9）。

【0 1 5 1】

制御機能 4 は、画面表示トランザクション情報オブジェクト 9 6 がセットされると、画面表示トランザクション情報オブジェクト 9 6 からレイアウト情報、動的表示情報、活性データに基づき、画面の状態をトランザクションが開始された時点の状態を含んだ任意の状態へ復帰する（2 0 1 0）。または、制御機能 4 は、画面表示トランザクション情報オブジェクト 9 6 を示す I D がセットされると

、IDから画面表示トランザクション情報オブジェクト96を検索し、以下同様の手順で画面の状態を復帰する。

【0152】

以上説明したように、本実施の形態に係る画面制御プログラム1を利用することにより、開発者は、表示を要求する時点毎に動的に変化する表示内容についても、この動的な表示内容を取得するために予め用意されている動的表示処理を指定し、またこの動的な表示内容を得るために利用される属性を指定するのみで、動的な表示内容を画面上に表示させるプログラムを作成できる。

【0153】

したがって、表示内容が変化する画面プログラムを開発・変更する場合であってもプログラムを記述する必要がなく、コンパイルする必要もない。

【0154】

ゆえに、開発者は、画面プログラムを容易かつ効率的に開発することができ、開発労力を軽減できる。

【0155】

また、本実施の形態に係る画面制御プログラム1を利用することにより、正常な入力内容を保持しておき、アプリケーション11が障害から復帰した場合に、画面に障害発生前の正常な入力状態を再現することができる。

【0156】

したがって、ユーザは画面に再入力する必要がなく、入力操作を効率化できる。また、開発者はこのような入力操作を効率化させる画面プログラムを容易に開発できる。

【0157】

なお、本実施の形態においては、画面制御プログラム1が画面表示トランザクション機能7を備えている場合について説明している。しかしながら、画面表示トランザクション機能7の実装形態は、この手法に限定されるものではない。例えば、画面表示トランザクション機能を実現するためのコンポーネントをプラグインする形態を採用してもよい。

【0158】

また、本実施の形態においてレイアウト情報取得機能 2 と動的表示情報取得機能 3 とは、画面を表示する際にそれぞれがレイアウト情報 9 と動的表示情報 1 0 とを取得し、変換し、変換結果を制御機能 4 に提供する。しかしながら、これに代えて、レイアウト情報取得機能 2 及び動的表示情報取得機能 3 は、それぞれレイアウト情報 9 及び動的表示情報 1 0 を予め取得、変換しておき、画面を表示する際に制御機能 4 がこの変換済みの情報を読み込むとしてもよい。

#### 【 0 1 5 9 】

##### （第 2 の実施の形態）

本実施の形態においては、レイアウト情報のセットを複数回に分けて行う画面制御プログラムについて説明する。

#### 【 0 1 6 0 】

図 2 1 は、本実施の形態に係る画面制御プログラムが計算機上で実現する機能を示すブロック図である。

#### 【 0 1 6 1 】

この画面制御プログラム 1 7 におけるレイアウト情報取得機能 1 8 は、レイアウト情報のセットを複数回に分けて行う場合に対応している。このレイアウト情報取得機能 1 8 は、複数に分割されたレイアウト情報 1 9 を入力する。

#### 【 0 1 6 2 】

図 2 2 は、本実施の形態に係る画面制御プログラム 1 7 がレイアウト情報を読み込んだ場合の処理手順を例示するシーケンス図である。

#### 【 0 1 6 3 】

処理 2 2 0 1 ～ 2 2 0 5 は、先の図 3 における処理 3 0 1 ～ 3 0 5 と同様である。これらの処理により制御機能 4 に対してレイアウト情報が設定され、画面にはフィールドが初期状態で表示される。

#### 【 0 1 6 4 】

次に、分割された別のレイアウト情報を指定してレイアウト情報の追加指示が、アプリケーション 1 1 からレイアウト情報取得機能 1 8 に対して行われる（2 2 0 6）。

#### 【 0 1 6 5 】

レイアウト情報取得機能 1 8 は、処理 2 2 0 2 と同様の手順でフィールドの属性情報を展開し（2 2 0 7）、制御機能 4 に対してレイアウト情報の追加を指示する（2 2 0 8）。

【0 1 6 6】

制御機能 4 は、追加されたフィールドのオブジェクトを作成し、属性をセットする（2 2 0 9）。以下、処理 2 2 0 7～2 2 0 8 が追加されたフィールドの数だけ繰り返される（2 2 1 0）。以上により必要に応じたレイアウトの初期化が可能になる。

【0 1 6 7】

なお、動的表示情報取得機能 3 も、このレイアウト情報取得機能 1 8 と同様に、動的表示情報 1 0 を複数回に分けて読み込み可能としてもよい。

【0 1 6 8】

本実施の形態に係る画面制御プログラム 1 7 を利用しても、上記第 1 の実施の形態と同様の作用効果を得ることができる。

【0 1 6 9】

（第 3 の実施の形態）

本実施の形態においては、レイアウト情報、動的表示情報がネットワーク上に配置された環境で利用される画面制御プログラムについて説明する。

【0 1 7 0】

図 2 3 は、本実施の形態に係る画面制御プログラムが計算機システム上で実現する機能を例示するブロック図である。

【0 1 7 1】

画面制御プログラム 2 0 を備える計算機 2 1 は、ネットワーク 2 2 を介して WWサーバ 2 3 と接続されている。

【0 1 7 2】

レイアウト情報取得機能 2 4 は、レイアウト情報 9 をネットワーク 2 2 を経由してダウンロードする。

【0 1 7 3】

動的表示情報取得機能 2 5 は、動的表示情報 1 0 をネットワーク 2 2 を経由し



てダウンロードする。

【0174】

WWWサーバ23は、レイアウト情報9、動的表示情報10のダウンロードを可能とするサーバプロセスの具体例である。

【0175】

レイアウト情報9をネットワーク22を経由してダウンロードする場合には、先の図3に示す処理301においてレイアウト情報9のURI (Uniform Resource Identifiers) を指定する。

【0176】

レイアウト情報取得機能24は、WWWサーバ23にアクセスして指定されたURIへのコネクションを確立し、レイアウト情報9を読み込む。

【0177】

動的表示情報10をネットワーク22を経由してダウンロードする場合もレイアウト情報9の場合と同様に行われ、先の図10に示す処理1001において動的表示情報10のURIを指定する。

【0178】

Java言語ではURIで指定されたファイルを扱うクラスが用意されている。このクラスを用いることで上記の動作は、実現可能である。

【0179】

本実施の形態に係る画面制御プログラム20を利用しても、上記第1の実施の形態と同様の作用効果を得ることができる。

【0180】

(第4の実施の形態)

本実施の形態においては、各インタフェースにプラグインされるコンポーネントをネットワーク上に分散させた環境で利用される画面制御プログラムについて説明する。

【0181】

図24は、本実施の形態に係る画面制御プログラムが計算機システム上で実現する機能を示すブロック図である。

## 【 0 1 8 2 】

画面制御プログラム 2 6 は、計算機上で動作する。画面制御プログラム 2 6 における検索処理インタフェース 2 7 は、検索処理スタブ 2 8 と接続されている。

## 【 0 1 8 3 】

この検索処理スタブ 2 8 は、検索処理インタフェース 2 7 にプラグインされる検索処理用のコンポーネントである。検索処理スタブ 2 8 は、ネットワーク 2 9 を介して検索処理リモートオブジェクト 3 0 に検索処理を依頼し、検索処理リモートオブジェクト 3 0 が行った検索処理結果を受け取り、検索処理インタフェース 2 7 に返却する。

## 【 0 1 8 4 】

同様に、入力チェック処理インタフェース 3 1 は、入力チェック処理スタブ 3 2 と接続されている。

## 【 0 1 8 5 】

この入力チェック処理スタブ 3 2 は、入力チェック処理インタフェース 3 1 にプラグインされる入力チェック処理用のコンポーネントである。入力チェック処理スタブ 3 2 は、ネットワーク 2 9 を介して入力チェック処理リモートオブジェクト 3 3 に入力チェック処理を依頼し、入力チェック処理リモートオブジェクト 3 3 が行った入力チェック処理結果を受け取り、入力チェック処理インタフェース 3 1 に返却する。

## 【 0 1 8 6 】

本実施の形態に係る画面制御プログラム 2 6 を利用しても、上記第 1 の実施の形態と同様の作用効果を得ることができる。

## 【 0 1 8 7 】

## (第 5 の実施の形態)

本実施の形態においては、画面表示トランザクション情報がサーバに保持され、サーバがクライアントの画面状態を制御するために利用される画面制御プログラムについて説明する。

## 【 0 1 8 8 】

図 2 5 は、本実施の形態に係る画面制御プログラムが計算機システム上で実現

する機能を示すブロック図である。

【 0 1 8 9 】

画面制御プログラム 3 4 は、クライアント上で動作する。画面制御プログラム 3 4 における画面表示トランザクション機能 3 5 は、画面表示トランザクション情報送受信部 3 6 に接続されている。

【 0 1 9 0 】

画面表示トランザクション情報送受信部 3 6 は、画面表示トランザクション機能 3 5 から画面表示トランザクション情報（画面の状態情報）を取得し、ネットワーク 3 7 を介して画面表示トランザクション管理サーバ 3 8 に送信する。

【 0 1 9 1 】

また、画面表示トランザクション情報送受信部 3 6 は、画面表示トランザクション管理サーバ 3 8 から受信した画面表示トランザクション情報を画面表示トランザクション機能 3 5 にセットする。

【 0 1 9 2 】

画面表示トランザクション管理サーバ 3 8 は、画面表示トランザクション送受信部 3 6 から受信した画面表示トランザクション情報をクライアント毎に管理する。

【 0 1 9 3 】

そして、画面表示トランザクション管理サーバ 3 8 は、管理している画面表示トランザクション情報を必要に応じて取り出し、画面表示トランザクション情報送受信部 3 6 に送信する。

【 0 1 9 4 】

本実施の形態に係る画面制御プログラム 3 4 を利用しても、上記第 1 の実施の形態と同様の作用効果を得ることができる。

【 0 1 9 5 】

（第 6 の実施の形態）

本実施の形態においては、業務処理をプラグインする画面制御プログラムについて説明する。

【 0 1 9 6 】

図 2 6 は、本実施の形態に係る画面制御プログラムが計算機上で実現する機能を示すブロック図である。

【 0 1 9 7 】

画面制御プログラム 3 9 の制御機能 4 0 は、業務処理インタフェース 4 1 と接続されている。

【 0 1 9 8 】

業務処理用のコンポーネント（各種のプログラム） 4 2 は、他の画面表示に関わるコンポーネントと同様の手法により、業務処理インタフェース 4 1 にプラグインされる。

【 0 1 9 9 】

アプリケーション 1 1 は、画面表示に関する処理のみではなく、業務処理もこの画面制御プログラム 3 9 を利用して実行可能である。

【 0 2 0 0 】

開発者は、この業務処理用のコンポーネント 4 2 の利用を望む場合には、他の画面表示に関わるコンポーネントと同様に、コンポーネント 4 2 を指定し、必要な属性を指定するのみでよい。

【 0 2 0 1 】

これにより、画面プログラムの開発のみではなく、アプリケーションの開発も効率化できる。

【 0 2 0 2 】

なお、上記各実施の形態で説明した発明は、それぞれを自由に組み合わせることができる。また、上記各実施の形態で説明した各機能、各要素は、同様の作用・機能を実現可能であれば配置を変更させてもよく、また各機能、各要素を自由に組み合わせてもよい。

【 0 2 0 3 】

上記各実施の形態で説明した画面制御プログラムの各機能、各要素は、コンピュータに実行させることのできるプログラムである。このプログラムは、例えば磁気ディスク（フロッピーディスク、ハードディスク等）、光ディスク（CD-ROM、DVD等）、半導体メモリなどの記録媒体 9 7 に書き込んでコンピュー

タに適用可能である。また、このプログラムは通信媒体により伝送して、計算機、計算機システムに適用可能である。

#### 【 0 2 0 4 】

上記各実施の形態で説明した画面制御プログラムの各機能を実現するコンピュータは、記録媒体に記録されたプログラムを読み込み、プログラムによって動作が制御されることにより、上述した処理を実行する。

#### 【 0 2 0 5 】

##### （第 7 の実施の形態）

本実施の形態においては、動的表示処理識別情報及び表示属性情報の入力により画面が実際にどのように変更されるかを説明する。

#### 【 0 2 0 6 】

図 2 7 は、変更前の画面の構成状態を例示する図である。

#### 【 0 2 0 7 】

この画面 4 3 の構成要素である「国」フィールド 4 4 及び「振り込み先」フィールド 4 5 に対しては、プルダウンメニューを利用可能とする表示属性が初期設定されている。

#### 【 0 2 0 8 】

一方、「取引相手」フィールド 4 6 には、ポップアップ検索を利用可能とする表示属性が初期設定されている。

#### 【 0 2 0 9 】

図 2 8 は、動的表示処理識別情報及び表示属性情報の入力による変更後の画面の構成状態を例示する図である。

#### 【 0 2 1 0 】

この画面 4 7 においては、「国」フィールド 4 4 に対してコンボボックスを利用可能とする動的表示属性が設定されている。さらに、画面 4 7 においては、新たに検索処理用のコンポーネント 8 a と入力チェック処理用のコンポーネント 8 b がプラグインされている。

#### 【 0 2 1 1 】

開発者は、「国」フィールド 4 4 に対して検索処理用のコンポーネント 8 a と

入力チェック処理用のコンポーネント 8 b をプラグインする旨を記述した動的表示処理識別情報を入力する。また、開発者は、入力内容が四桁の数字と 1 以上の文字であるか否かをチェックする旨を記述した表示属性情報（動的表示属性情報）を入力する。

#### 【 0 2 1 2 】

これにより、「国」フィールド 4 4 の入力チェックを容易に行うことができる。

#### 【 0 2 1 3 】

また、「国」フィールド 4 4 の一覧表示ボタン 9 8 を押すことで、データベースが検索され、図 2 9 に示すような検索結果の一覧 5 2 が表示される。そして、一覧に対するユーザからの選択を受付可能な状態となる。

#### 【 0 2 1 4 】

画面 4 7 においては、同様の手法により、「振り込み先」フィールド 4 5 に対しても、入力チェック処理用のコンポーネント 8 b がプラグインされている。また、このコンポーネント 8 b に対する表示属性情報（動的表示属性情報）も入力されている。

#### 【 0 2 1 5 】

さらに、この「振り込み先」フィールド 4 5 に対しては、ポップアップ処理用のコンポーネント 8 c がプラグインされている。このポップアップ処理用のコンポーネント 8 c に対する表示属性情報では、ポップアップウィンドウに表示すべき画面を構成するクラスが指定されている。

#### 【 0 2 1 6 】

さらに、この画面 4 7 の「取引相手」フィールド 4 6 には、ポップアップ検索を利用可能とする表示属性に代えてプルダウンメニューを利用可能とする表示属性が設定されている。また、このフィールド 4 6 に対して検索処理用のコンポーネント 8 a もプラグインされている。

#### 【 0 2 1 7 】

さらに、画面 4 7 上には、「設備費別」フィールド 4 8 に対して、上記「取引相手」フィールド 4 6 と同様に、プルダウンメニューを利用可能とする表示属性

が設定されており、検索処理用のコンポーネント 8 a がプラグインされている。  
また、この画面 4 7 には、「計上年月」フィールド 4 9、「年」フィールド 5 0 が追加されている。これらのフィールド 4 9、5 0 には、テキストフィールドとして利用可能とする表示属性が設定されている。

## 【 0 2 1 8 】

図 3 0 は、「振り込み先」フィールド 4 5 に対してプラグインされたポップアップ処理用のコンポーネント 8 c が起動された画面の状態を例示する図である。

## 【 0 2 1 9 】

画面 5 3 において、「振り込み先」フィールド 4 5 の検索ボタン 9 9 が押されると、ポップアップウィンドウ 5 4 が表示される。このポップアップウィンドウ 5 4 に検索結果が表示される。

## 【 0 2 2 0 】

図 3 1 は、検索結果の選択を行った画面 5 3 の状態を例示する図である。

## 【 0 2 2 1 】

検索された結果のうちのいずれかの要素がユーザによって選択されると、ポップアップウィンドウ 5 4 は閉じられ、自動的に「振り込み先」フィールド 4 5 に選択された値がセットされる。

## 【 0 2 2 2 】

## (第 8 の実施の形態)

本実施の形態においては、画面を構成する G U I コンポーネント間でデータを交換するためのインタフェースを定義する。

## 【 0 2 2 3 】

また、本実施の形態においては、G U I コンポーネントとこの G U I コンポーネントに付加されるプログラムやデータなどのカスタムコンポーネントとの間で、データを交換するためのインタフェースを定義する。

## 【 0 2 2 4 】

なお、G U I コンポーネントとは、ユーザ・インタフェースの実現するための画面コンポーネントである。

## 【 0 2 2 5 】

図 3 2 は、本実施の形態に係る G U I コンポーネントのインタフェース機能によるデータ交換状態を例示するブロック図である。

【 0 2 2 6 】

画面を構成する G U I コンポーネントは、アプリケーションに依存しない部分のプログラムを持つ。

【 0 2 2 7 】

また、G U I コンポーネントは、階層的な関係を持つ。管理型コンポーネントは、他の G U I コンポーネントの親として動作する。被管理型コンポーネントは、管理型コンポーネントの上で動作する。

【 0 2 2 8 】

この図 3 2 において、パネル 5 5 は管理型コンポーネントである。テキストフィールド 5 6、コンボボックス 5 7、ボタン 5 8 は被管理型コンポーネントである。

【 0 2 2 9 】

本実施の形態では、G U I コンポーネント間のインタフェース、及び G U I コンポーネントとカスタムコンポーネントとの間のインタフェースとを予め定義している。

【 0 2 3 0 】

すなわち、管理型コンポーネント及び被管理型コンポーネントは、自身に装着されるカスタムコンポーネントとの間でデータ交換を行うためのインタフェース機能 5 9 を持つ。また、管理型コンポーネント及び被管理型コンポーネントは、他の G U I コンポーネントとの間でデータ交換を行うためのインタフェース機能 6 0 を持つ。

【 0 2 3 1 】

さらに、管理型コンポーネント及び被管理型コンポーネントにおいては、例えば、キー入力されたら入力チェック処理の実行を依頼する、一覧を取得する条件が設定されたら検索を依頼し検索結果を受け取って表示する、ボタンが押されたらプログラムの呼び出しを行う、などのような G U I コンポーネントの基本動作部分が、フレームワークとして予め定義されている。これにより、G U I コンポ



ーネットの基本動作部分は、GUIコンポーネントの一部となる。

【0232】

GUIコンポーネントに装着されるカスタムコンポーネントは、アプリケーションに依存する部分のプログラムやデータを含んでいる。このアプリケーションに依存するプログラムには、例えば入力チェック処理用プログラム61a、検索処理用プログラム61b、データ収集処理用プログラム61cなどがある。

【0233】

この画面において、入力チェック処理用プログラム61aは、テキストフィールド56からインタフェース機能59を利用して入力チェック処理の実行依頼を受け取り、入力チェック処理を実行し、その結果に応じた動作を実行する。

【0234】

検索処理用プログラム61bは、コンボボックス57からインタフェース機能60を利用して検索依頼を受け取り、この依頼の内容に基づいて所定のデータベース62に対する検索を実行し、インタフェース機能59を利用して検索結果をコンボボックス57に返す。

【0235】

データ収集処理用プログラム61cは、呼び出されるとインタフェース機能59を利用してパネル55上の入力内容を取得する。

【0236】

データ61dは、ボタン58が押された場合にインタフェース機能59を利用してアプリケーション11に読み込まれる。

【0237】

例えば、ボタン58が押された場合、このボタン58は、インタフェース機能59を利用して自己に装着されているデータ61dを読み込む。データ61dでは、データ収集処理用プログラム61cのメソッドを指定している。

【0238】

すると、ボタン58は、GUIコンポーネント間のインタフェース機能60を利用してデータ収集処理用プログラム61cを呼び出す。

【0239】

データ収集処理プログラム 6 1 c は、呼び出されると、インタフェース機能 5 9 及びインタフェース機能 6 0 を利用し、テキストフィールド 5 6 に入力されているデータを取得する。さらに、データ収集処理プログラム 6 1 c は、呼び出されると、インタフェース機能 5 9 及びインタフェース機能 6 0 を利用し、コンボボックス 5 7 に入力されているデータを取得する。

【 0 2 4 0 】

そして、データ収集処理プログラム 6 1 c は、収集したデータをアプリケーション 1 1 に送信する。

【 0 2 4 1 】

ここで、本実施の形態における G U I コンポーネントとカスタムコンポーネントとを利用して画面プログラムを作成する手順を以下に示す。

【 0 2 4 2 】

まず、画面作成ツールを起動する。起動された画面作成ツールは、図 3 3 に示すような初期状態の画面を表示する。

【 0 2 4 3 】

例えば、開発者は、図 3 4 に示すように、パネル 5 5 を配置する操作を行う。そして、図 3 5 に示すように、パネル 5 5 上にテキストフィールド 5 6 を配置する。

【 0 2 4 4 】

すると、テキストフィールド 5 6 に装着可能なカスタムコンポーネントの一覧を列挙したウィンドウ 6 3 が表示される。

【 0 2 4 5 】

ここで、開発者は、このウィンドウ 6 3 上で入力チェック処理用プログラム 6 1 a を指定したとする。

【 0 2 4 6 】

すると、このテキストフィールド 5 6 のプロパティとして、入力チェック処理用プログラムの属性を設定するためのウィンドウ 6 4 が表示される。

【 0 2 4 7 】

開発者は、このウィンドウ 6 4 上で入力チェック結果に応じた動作の属性を設

定する。

【 0 2 4 8 】

以上のように、本実施の形態においては、G U I コンポーネントとカスタムコンポーネントとを組み合わせる画面プログラムを作成する手法を実現している。

【 0 2 4 9 】

これにより、開発者は、入力チェックのルール及びチェック結果に応じた動作、検索条件などのようなアプリケーションに依存する部分だけを属性として指定するのみで、様々な動作を行う画面プログラムを作成できる。

【 0 2 5 0 】

したがって、開発者の作業を軽減できる。また、インタフェースが定められているため、画面を構成する部分を複数の開発者で分担し、後で組み合わせることができる。

【 0 2 5 1 】

(第 9 の実施の形態)

本実施の形態においては、上記第 8 の実施の形態に係る G U I コンポーネントとカスタムコンポーネントとについて具体的に説明する。

【 0 2 5 2 】

図 3 6 は、G U I コンポーネントとカスタムコンポーネントとの構成を示すクラス図である。

【 0 2 5 3 】

全ての G U I コンポーネントは、共通に扱えるデータ型である「G U I コンポーネントテキスト 6 5」を持つ。

【 0 2 5 4 】

全ての G U I コンポーネントは、「G U I コンポーネントコンテキスト 6 5」をやり取り可能なインタフェース「G U I コンポーネントインタフェース 6 6」にしたがって作成される。

【 0 2 5 5 】

G U I コンポーネントには、自らの上に他の G U I コンポーネントを配置可能なコンポーネントがあり、管理型コンポーネントと定義する。このコンポーネン

トは、配置されたコンポーネントを管理可能である。管理型コンポーネントは、自らに部品を配置し管理するための「管理型コンポーネントインタフェース 6 7」にしたがって作成される。このインタフェースにしたがって実際に動作する G U I コンポーネントとして作成されたコンポーネントが「管理型ベースコンポーネント 6 8」である。これは具体的にはパネルのような G U I コンポーネントである。管理型コンポーネントは、他の管理型コンポーネントも管理可能である。

## 【 0 2 5 6 】

一方、他のコンポーネントを管理しないコンポーネントを、被管理型コンポーネントと定義する。この被管理型コンポーネントは、管理型コンポーネントに管理されるための「被管理型コンポーネントインタフェース 6 9」にしたがって作成される。このインタフェースにしたがって実際に動作する G U I コンポーネントとして作成されたものが「被管理型ベースコンポーネント 7 0」である。

## 【 0 2 5 7 】

以上が、アプリケーションに依存しない基本動作部分を提供するコンポーネントである。

## 【 0 2 5 8 】

アプリケーションに依存する部分は、カスタムコンポーネントと称する着脱可能なプログラムとして定義し、基本動作を行うベースコンポーネントに装着する。この装着により、基本動作に加えてアプリケーションに依存した動作が加わる。

## 【 0 2 5 9 】

ベースコンポーネントとカスタムコンポーネントは、その性質により複数種類に分類されており、ベースコンポーネントとカスタムコンポーネントの対応関係は予め規定されている。

## 【 0 2 6 0 】

管理型コンポーネントには、1 種類以上のカスタムコンポーネントを装着することが可能である。カスタムコンポーネントの種類によって、ベースコンポーネントから起動されるインタフェースは決まっている。ベースコンポーネントから起動されるインタフェースを図 3 6 では「カスタムコンポーネント 1 インタフェ

ース 7 1」としている。このインタフェースにしたがって具体的な動作を記述したプログラム 7 2 を作成する。

#### 【 0 2 6 1 】

同様に、被管理型コンポーネントにも 1 種類以上のカスタムコンポーネントを装着することが可能である。ベースコンポーネントからカスタムコンポーネントを呼び出すインタフェースは決まっている。このインタフェースを図 3 6 では「カスタムコンポーネント 2 インタフェース 7 3」としている。このインタフェースにしたがって具体的な動作を記述したプログラム 7 4 を作成する。

#### 【 0 2 6 2 】

図 3 7 に、管理型コンポーネントと被管理型コンポーネントの関係を示す。

#### 【 0 2 6 3 】

管理型コンポーネントインタフェース 6 7 は、管理型コンポーネントが持つべきインタフェースである。J a v a 言語では `interface` という形態で定義される。Interface には外部に対してどんな操作を提供すべきか定義されている。管理型コンポーネントインタフェース 6 7 には、コンポーネントを配置し管理対象に加えるための操作、自分自身の I D を設定／取得するための操作、自分自身が持つコンテキストを設定／取得するための操作、管理対象のコンポーネントをそのコンポーネントの I D をキーにして取得するための操作、管理対象のコンポーネントが持つコンテキストを取得するための操作が定義されている。

#### 【 0 2 6 4 】

一方、被管理型コンポーネントインタフェース 6 9 は、被管理型コンポーネントが持つべきインタフェースである。被管理型コンポーネントインタフェース 6 9 には、自分自身の I D を設定／取得するための操作、自分自身が持つコンテキストを取得するための操作が定義されている。

#### 【 0 2 6 5 】

管理型コンポーネントインタフェース 6 7 と被管理型コンポーネントインタフェース 6 9 の関係は 1 対多であり、また管理型コンポーネントインタフェース 6 7 は自分と同じタイプのコンポーネントも管理可能で、その関係も 1 対多である。

## 【 0 2 6 6 】

図 3 8 は、管理型コンポーネントと被管理型コンポーネントとの間の処理手順を例示するシーケンス図である。ここで説明する手順は、アプリケーションから画面上に表示されるデータ、またはユーザから入力／選択されたデータを収集する処理の手順である。アプリケーションが収集すべき画面の現在のデータは、先で述べた活性データである。

## 【 0 2 6 7 】

アプリケーション 7 5 は、本実施の形態において作成される画面プログラムを利用するアプリケーションである。

## 【 0 2 6 8 】

管理型コンポーネント 7 6 は、管理型コンポーネントのインスタンスである。

## 【 0 2 6 9 】

被管理型コンポーネント 7 7 は、管理型コンポーネント 7 6 上に配置されて管理される G U I コンポーネントである。

## 【 0 2 7 0 】

コンテキスト 7 8 は、G U I コンポーネントが持つコンテキストである。

## 【 0 2 7 1 】

アプリケーション 7 5 から管理型コンポーネント 7 6 に対し、活性データを取得する処理の呼び出しが行われる（3 8 0 1）。

## 【 0 2 7 2 】

すると、管理型コンポーネント 7 6 は自分が管理している G U I コンポーネントの一覧を取得する（3 8 0 2）。

## 【 0 2 7 3 】

次に、取得した一覧に格納されている G U I コンポーネントに対し、コンテキストを取得する処理の呼び出しが行われる（3 8 0 3）。G U I コンポーネント 7 7 は、自分のコンテキストを戻り値として返す（3 8 0 4）。

## 【 0 2 7 4 】

コンテキストには、その G U I コンポーネントの活性データが格納されているので、管理型コンポーネント 7 6 は得られたコンテキストに対し、活性データを

取得する処理の呼び出しを行う（3805）。

【0275】

コンテキスト78は、活性データを戻り値として返す（3806）。

【0276】

管理型コンポーネント76は、自分が管理しているGUIコンポーネント分、上記処理を繰り返す。そして、得られた活性データをコンポーネントのIDと組にして一覧形式にし、戻り値としてアプリケーションに返す（3807）。

【0277】

これにより、アプリケーションの開発者は、どんな種類のGUI部品が配置されているか、GUI部品毎に値を取り出すにはどのような処理を呼び出すべきかを考慮することなく画面データを取得可能である。

【0278】

次に、図39～図44に、ベースコンポーネントとカスタムコンポーネントとの関係を示す。

【0279】

まず、管理型コンポーネントの一種であるパネルコンポーネントの例を示す。

【0280】

図39は、パネルコンポーネントにおけるベースコンポーネントとカスタムコンポーネントの関係を示している。

【0281】

パネルコンポーネント79が外部に提供すべき操作は、パネル型ベースコンポーネントインタフェース80として定義されている。実際に動作するパネルコンポーネント79は、パネル型ベースコンポーネントインタフェース80を実装して作成される。ベースコンポーネントのインタフェースには、どんなカスタムコンポーネントが装着されるかも定義されている。パネル型ベースコンポーネントインタフェース80には、決定実行型カスタムコンポーネントインタフェース81と一覧検索型カスタムコンポーネントインタフェース82が装着される。

【0282】

カスタムコンポーネントの具体的な動作は、決定実行型カスタムコンポーネン

トインタフェース 8 1 を実装して決定実行型カスタムコンポーネント 8 3 が作成され、一覧検索型カスタムコンポーネントインタフェース 8 2 を実装して一覧検索型カスタムコンポーネント 8 4 が作成される。

## 【 0 2 8 3 】

図 4 0 ～図 4 2 は、パネルコンポーネントにおけるベースコンポーネントとカスタムコンポーネントとの間の処理手順を示す。

## 【 0 2 8 4 】

図 4 0 は、アプリケーションからパネルコンポーネントに対して決定操作がなされた時の処理を示すシーケンス図である。

## 【 0 2 8 5 】

アプリケーション 7 5 からパネルコンポーネント 7 9 に決定操作の呼び出しが行われる ( 4 0 0 1 ) 。

## 【 0 2 8 6 】

パネルコンポーネント 7 9 は、自分自身に決定実行型カスタムコンポーネント 8 3 が設定されているか否かを判断し、設定されている場合は決定実行型カスタムコンポーネント 8 3 の実行操作を呼び出す ( 4 0 0 2 ) 。

## 【 0 2 8 7 】

図 4 1 は、ユーザによるボタン操作をきっかけとしてパネルコンポーネント 7 9 の決定操作が行われた場合の処理を示すシーケンス図である。

## 【 0 2 8 8 】

ボタンコンポーネント 8 5 は、パネルコンポーネント 7 9 上に配置されたコンポーネントである。

## 【 0 2 8 9 】

アプリケーションを操作するユーザがパネルコンポーネント 8 5 上に配置されたボタンコンポーネント 7 9 を押下する ( 4 1 0 1 ) 。ボタンコンポーネント 8 5 は、自分の配置されているパネルコンポーネント 7 9 の決定操作を呼び出す ( 4 1 0 2 ) 。パネルコンポーネント 6 0 3 は自分自身に決定実行型カスタムコンポーネントが設定されているか否かを判断し、設定されている場合は決定実行型カスタムコンポーネント 8 3 の実行操作を呼び出す ( 4 1 0 3 ) 。



## 【 0 2 9 0 】

図 4 2 は、パネルコンポーネント 7 9 と一覧検索型カスタムコンポーネント 8 4 との間の処理手順を示したシーケンス図である。

## 【 0 2 9 1 】

検索条件オブジェクト 8 6 は、一覧データを検索するための検索条件を格納するオブジェクトである。

## 【 0 2 9 2 】

一覧表示コンポーネント 8 7 は、パネルコンポーネント 7 9 上に配置された一覧データを表示するタイプの G U I コンポーネントである。

## 【 0 2 9 3 】

アプリケーション 7 5 で一覧検索するための条件を格納する検索条件オブジェクト 8 6 が作成される ( 4 2 0 1 ) 。この検索条件オブジェクト 8 6 は検索条件が与えられると生成される。検索条件オブジェクト 8 6 の生成は、一覧データを作成したいコンポーネント分、繰り返される。

## 【 0 2 9 4 】

アプリケーション 7 5 は、こうして得られた検索条件をコンポーネントの I D と組にしてパネルコンポーネント 7 9 に設定する ( 4 2 0 2 ) 。

## 【 0 2 9 5 】

パネルコンポーネント 7 9 は、一覧検索型カスタムコンポーネント 8 4 が自分自身に設定されているか否かを判別し、設定されていたら検索条件オブジェクト 8 6 を渡して一覧検索型カスタムコンポーネント 8 4 の検索操作を呼び出す ( 4 2 0 3 ) 。

## 【 0 2 9 6 】

一覧検索型カスタムコンポーネント 8 4 は、与えられた検索条件にしたがってデータベースやファイルの検索処理を行い、検索結果を戻り値としてパネルコンポーネント 7 9 に返す ( 4 2 0 4 ) 。

## 【 0 2 9 7 】

パネルコンポーネント 7 9 は、得られた検索結果を I D で対応関係が定義された一覧表示コンポーネント 8 7 に設定する。そして、処理 4 2 0 2 ~ 4 2 0 4 を

与えられた条件の組の数だけ繰り返す（4 2 0 5）。

## 【0 2 9 8】

次に、被管理型コンポーネントの例を示す。被管理型コンポーネントには、その用途によっていくつかの種類があり、それぞれの種類のベースコンポーネントとカスタムコンポーネントの関係が決められている。その例が以下の表 2 である。ここでは、テキストフィールドコンポーネントの例を示す。

## 【0 2 9 9】

【表 2】

表 2

コンポーネントの 種別	コンポーネントの 実装	データの set/get	入力 チェック	一覧 データの set/get	一覧 検索	submit	popup
パネル	パネル	✓	-	✓	✓	✓	-
ラジオボタン パネル	ラジオボタン パネル	✓	-	-	-	-	-
実行コンポーネント	ボタン	✓	-	-	-	✓	✓
表示コンポーネント	ラベル	✓	-	-	-	-	-
テキスト入力 コンポーネント	テキストフィールド	✓	✓	-	-	-	-
	パスワード フィールド	✓	-	-	-	-	-
	テキストエリア	✓	✓	-	-	-	-
	HTML 表示エリア	✓	✓	-	-	-	-
固定選択 コンポーネント	チェックボックス	✓	-	-	-	-	-
	ラジオボタン	✓	-	-	-	-	-
	トグルボタン	✓	-	-	-	-	-
一覧単一選択 コンポーネント	コンボボックス	✓	✓	✓	-	-	-
一覧複数選択 コンポーネント	リスト	✓	-	✓	-	-	-
連続値 コンポーネント	プログレスバー	✓	-	-	-	-	-
	スライダー	✓	-	-	-	-	-
複合モデル コンポーネント	テーブル	✓	-	-	-	-	-
	ツリー	✓	-	-	-	-	-

## 【0 3 0 0】

図 4 3 は、テキストフィールドコンポーネントにおけるベースコンポーネントとカスタムコンポーネントの関係を示したシーケンス図である。

## 【0 3 0 1】

テキストフィールドコンポーネント 8 8 が外部に提供すべき操作は、テキスト

入力型ベースコンポーネントインタフェース 8 9 として定義されている。実際に動作するテキストフィールドコンポーネント 8 8 は、テキスト入力型ベースコンポーネントインタフェース 8 9 を実装して作成される。ベースコンポーネントのインタフェースには、どんなカスタムコンポーネントが装着されるかも定義されている。テキスト入力型ベースコンポーネントインタフェース 8 9 には、評価型カスタムコンポーネントインタフェース 9 0 と実行型カスタムコンポーネントインタフェース 9 1 が装着される。カスタムコンポーネントの具体的な動作を定めるために、評価型カスタムコンポーネント 9 1 は、評価型カスタムコンポーネントインタフェース 9 0 を実装して作成される。また、実行型カスタムコンポーネント 9 3 は、実行型カスタムコンポーネントインタフェース 9 1 を実装して作成される。

#### 【 0 3 0 2 】

図 4 4 は、テキストフィールドコンポーネント 8 8 におけるベースコンポーネントとカスタムコンポーネント間の処理手順を示すシーケンス図である。

#### 【 0 3 0 3 】

評価型カスタムコンポーネント 9 2 1 は、上記図 4 3 に示す評価型カスタムコンポーネント 9 2 のインスタンスの一つである。

#### 【 0 3 0 4 】

実行型カスタムコンポーネント 9 3 1 は、上記図 4 3 に示す実行型カスタムコンポーネント 9 3 のインスタンスの一つである。この実行型カスタムコンポーネント 9 3 1 は、評価型カスタムコンポーネント 9 2 1 と対応付けられている。

#### 【 0 3 0 5 】

評価型カスタムコンポーネント 9 2 2 は、上記図 4 3 に示す評価型カスタムコンポーネント 9 2 のインスタンスの一つである。この評価型カスタムコンポーネント 9 2 2 は、評価型カスタムコンポーネント 9 2 1 とは別のコンポーネントである。

#### 【 0 3 0 6 】

実行型カスタムコンポーネント 9 3 2 は、上記図 4 3 に示す実行型カスタムコンポーネント 9 3 のインスタンスの一つである。この実行型カスタムコンポーネ

ント 9 3 2 は、実行型カスタムコンポーネント 9 3 1 とは別のコンポーネントであり、評価型カスタムコンポーネント 9 2 2 と対応付けられている。

【 0 3 0 7 】

実行型カスタムコンポーネント 9 3 3 は、上記図 4 3 に示す実行型カスタムコンポーネント 9 3 のインスタンスの一つである。この実行型カスタムコンポーネント 9 3 3 は、他の実行型カスタムコンポーネント 9 3 1、9 3 2 とそれぞれ別のコンポーネントである。

【 0 3 0 8 】

ユーザは、テキストフィールドコンポーネント 8 8 にテキストを入力した後、入力チェックのトリガーとなるキーを打鍵する（4 4 0 1）。入力チェックのトリガーとなるキーは自由に設定可能である。

【 0 3 0 9 】

テキストフィールドコンポーネント 8 8 は、登録されている評価型カスタムコンポーネントのうち最初のコンポーネントである評価型カスタムコンポーネント 9 2 1 の入力値の評価操作を呼び出す（4 4 0 2）。

【 0 3 1 0 】

評価型カスタムコンポーネント 9 2 1 は、テキストフィールドコンポーネント 8 8 から渡された値を評価し、結果を戻り値として返す（4 4 0 3）。

【 0 3 1 1 】

もしここで、評価結果が偽であった場合、エラー処理を行うために評価型カスタムコンポーネント 9 2 1 と対応付けられており登録されている実行型カスタムコンポーネント 9 3 1 の実行操作を呼び出し、処理を終える（4 4 0 4）。

【 0 3 1 2 】

評価結果が真であった場合、次の評価を行うために、登録されている評価型カスタムコンポーネントのうち 2 番目のコンポーネントである評価型カスタムコンポーネント 9 2 2 の入力値の評価操作を呼び出す（4 4 0 5）。

【 0 3 1 3 】

評価型カスタムコンポーネント 9 2 2 は、テキストフィールドコンポーネント 8 8 から渡された値を評価し、結果を戻り値として返す（4 4 0 6）。

## 【 0 3 1 4 】

もしここで、評価結果が偽であった場合、エラー処理を行うために、評価型カスタムコンポーネント 9 2 2 と対応付けられ登録されている実行型カスタムコンポーネント 9 3 2 の実行操作を呼び出し、処理を終える（4 4 0 7）。

## 【 0 3 1 5 】

以上のように、全ての評価型カスタムコンポーネント 9 2 1、9 2 2 による評価を繰り返し、全ての評価結果が真だった場合、デフォルト動作として登録されている実行型カスタムコンポーネント 9 3 3 の実行操作が呼び出される（4 4 0 8）。

## 【 0 3 1 6 】

この処理は、以下のように、より具体的に説明できる。

## 【 0 3 1 7 】

上記図 4 4 の評価型カスタムコンポーネント 9 2 1 は、入力された値が半角数字か否かを判断するコンポーネントとする。

## 【 0 3 1 8 】

実行型カスタムコンポーネント 9 3 1 は、「半角数字を入力してください」というメッセージを表示するコンポーネントとする。

## 【 0 3 1 9 】

評価型カスタムコンポーネント 9 2 2 は、入力された値が所定の桁数以下であるか否かを判断するコンポーネントであって、判断する桁数の属性値が 8 桁と設定されているとする。

## 【 0 3 2 0 】

実行型カスタムコンポーネント 9 3 2 は、「8 桁以下の数値を入力してください」というメッセージを表示するコンポーネントとする。

## 【 0 3 2 1 】

実行型カスタムコンポーネント 9 3 3 は、入力された値にカンマ編集を施して返すコンポーネントであるとする。

## 【 0 3 2 2 】

また、入力チェックのトリガーとして Enter キーが指定されていたとする。

## 【 0 3 2 3 】

この場合、ユーザがテキストフィールドコンポーネント 8 8 にテキストを入力してEnterキーを打鍵すると、例えば「123456787」のように、入力された値が8桁以下の半角数字の場合には、カンマ編集された値「12,345,678」が表示される。

## 【 0 3 2 4 】

例えば「a1234567」のように、入力された値が半角数字以外の文字を含んだ値の場合には、「半角数字を入力してください」というメッセージが表示される。

## 【 0 3 2 5 】

例えば「123456789」のように、入力された値が半角数字のみからなるが8桁以上の場合には、「8桁以下の数値を入力してください」というメッセージが表示される。

## 【 0 3 2 6 】

以上説明したように、本実施の形態においては、アプリケーションに依存しない共通の性質に基づく部分は、画面を構成するGUIコンポーネントに持たせている。その一方で、アプリケーションに依存し、ある程度変更が必要になる部分は、GUIコンポーネントに装着するとしている。そして、GUIコンポーネント間、及びGUIコンポーネントとこのGUIコンポーネントに装着されるカスタムコンポーネントとの間は、所定のインタフェースにしたがってデータを交換する。

## 【 0 3 2 7 】

これにより、画面プログラムの開発効率が飛躍的に向上する。具体的には、以下のような効果が得られる。

## 【 0 3 2 8 】

データベースやファイルから一覧データを検索して画面に表示するプログラムを作成する場合に、SQL文を書いて検索用クラスを作成し、検索用クラスを呼ぶコードを書き、検索結果をGUIコンポーネントにセットするコードを書く、という作業を削減できる。

## 【 0 3 2 9 】

また、ユーザのテキスト入力をチェックするプログラムを作成する場合に、ユーザのキー入力イベントを取得するコードを書き、ユーザのキー入力イベントによって起動されるクラスを作成し、そのクラスに入力チェックルーチンを書き、また入力チェック結果のメッセージ出力やフォーマット編集のコードを書く、という作業を削減できる。

#### 【0330】

ポップアップウィンドウを表示し、ポップアップウィンドウで入力された値をメイン画面のコンポーネントに反映するプログラムを作成する場合に、ボタンの押下イベントによりポップアップウィンドウを表示するコードを書き、ポップアップウィンドウで入力された値を取得するコードを書き、取得した値をメイン画面のコンポーネントに設定するコードを書く、という作業を削減できる。

#### 【0331】

アプリケーションに入力データを提供するためのプログラムを作成する場合に、画面に入力されたデータを個々のコンポーネントから収集し、それをまとめるというコードを書き、そのデータを業務ロジックに渡して呼び出すというコードを書く、という作業を削減できる。

#### 【0332】

画面プログラムを分割して作成する場合に、パネル経由でコンポーネントのデータにアクセスする統一された方法が提供されているため、パネルにコンポーネントが配置されただけでそのまま再利用可能となる。よって分散開発が容易になる。

#### 【0333】

##### 【発明の効果】

以上詳記したように本発明においては、動的に変化する表示内容を得るために予め用意されている動的表示処理を指定し、この動的表示処理で利用する属性を指定するのみで、動的な表示内容を取得し画面に表示させることができる。

#### 【0334】

したがって、画面プログラムの開発や変更においてプログラムを記述する必要がなく、コンパイルする必要もない。ゆえに、ユーザは画面プログラムを効率的

かつ容易に開発・変更でき、ユーザの労力を軽減できる。

【0335】

また、本発明においては、正常な入力を保持しておき、障害発生後にはこの保持しておいた内容に基づいて、入力内容を再現できる。

【0336】

したがって、ユーザの操作が効率化され、容易に画面プログラムの耐障害性を向上させることができる。

【0337】

さらに、本発明においては、画面を構成するために用意されている画面コンポーネント間のデータ交換のためのインタフェースを定義している。また、画面コンポーネントとこの画面コンポーネントに装着されるカスタムコンポーネントとの間のデータ交換のためのインタフェースを定義している。

【0338】

これにより、画面プログラムの開発が容易であり、分散して開発することも容易になる。

【図面の簡単な説明】

【図1】

本発明の第1の実施の形態に係る画面制御プログラムによって計算機上で実現される機能を示すブロック図。

【図2】

レイアウト情報の具体例を示す図。

【図3】

レイアウト情報が画面制御プログラムに読み込まれた場合の処理手順を例示する図。

【図4】

<field> タグの数だけ繰り返される各フィールドの作成処理を例示するフローチャート。

【図5】

レイアウト情報に基づく制御機能の処理によって構成される画面を例示する図



【図 6】

レイアウト情報が画面制御プログラムによって読み込まれた場合の処理手順の変形例を示す図。

【図 7】

レイアウト情報において定義された入力チェック属性にしたがって、画面制御プログラムによって実行される入力チェック処理の処理手順を例示するシーケンス図。

【図 8】

入力チェックの結果がエラーだった場合の画面を例示する図。

【図 9】

動的表示情報の具体例を示す図。

【図 1 0】

動的表示情報が画面制御プログラムによって読み込まれた場合の処理手順を例示するシーケンス図。

【図 1 1】

動的表示情報に基づく制御機能の処理によって構成される画面を例示する図。

【図 1 2】

動的表示情報が画面制御プログラムによって読み込まれた場合の処理手順の変形例を示すシーケンス図。

【図 1 3】

フィールドにセットされた表示属性が検索だった場合に画面制御プログラムによって実行される処理の手順を例示するシーケンス図。

【図 1 4】

フィールドにセットされた表示属性が検索だった場合の画面制御プログラムの処理を例示するフローチャート。

【図 1 5】

検索処理用のコンポーネントによって動的な表示内容が取得され、検索された一覧がユーザの操作に基づいて表示された画面を例示する図。

【図 1 6】

フィールドにセットされた表示属性がポップアップだった場合に画面制御プログラムによって実行される処理の手順を例示するシーケンス図。

【図 1 7】

ユーザによるフィールドへの操作によって起動されるポップアップ処理の手順を例示するシーケンス図。

【図 1 8】

ポップアップウィンドウが表示されている画面を示す図。

【図 1 9】

ポップアップウィンドウで選択されたデータがフィールドにセットされた画面を示す図。

【図 2 0】

画面表示トランザクション機能を利用して画面表示を復帰させる処理手順を例示するシーケンス図。

【図 2 1】

本発明の第 2 の実施の形態に係る画面制御プログラムによって計算機上で実現される機能を示すブロック図。

【図 2 2】

同実施の形態に係る画面制御プログラムによってレイアウト情報が読み込まれた場合の処理手順を例示するシーケンス図。

【図 2 3】

本発明の第 3 の実施の形態に係る画面制御プログラムによって計算機上で実現される機能を示すブロック図。

【図 2 4】

本発明の第 4 の実施の形態に係る画面制御プログラムによって計算機システム上で実現される機能を示すブロック図。

【図 2 5】

本発明の第 5 の実施の形態に係る画面制御プログラムによって計算機システム上で実現される機能を示すブロック図。

【図 2 6】

本発明の第 6 の実施の形態に係る画面制御プログラムが計算機上で実現する機能を示すブロック図。

【図 2 7】

本発明の第 7 の実施の形態における変更前の画面を示す図。

【図 2 8】

動的表示処理識別情報及び表示属性情報の入力による変更後の画面の構成状態を例示する図。

【図 2 9】

一覧表示ボタンが押された画面の状態を例示する図。

【図 3 0】

「振り込み先」フィールドに対してプラグインされたポップアップ処理用のコンポーネントが起動された画面の状態を例示する図。

【図 3 1】

検索結果の選択を行った画面の状態を例示する図。

【図 3 2】

本発明の第 8 の実施の形態に係る G U I コンポーネントのインタフェース機能によるデータ交換状態を例示するブロック図。

【図 3 3】

同実施の形態における画面作成ツールの初期状態の画面を示す図。

【図 3 4】

同実施の形態における画面作成ツールの画面上でパネルを配置した状態を示す図。

【図 3 5】

同実施の形態における画面作成ツールによりパネル上にテキストフィールドを配置した状態を示す図。

【図 3 6】

本発明の第 9 の実施の形態に係る G U I コンポーネントとカスタムコンポーネントとの構成を示すクラス図。

【図 3 7】

管理型コンポーネントと被管理型コンポーネントの関係を例示する図。

【図 3 8】

管理型コンポーネントと被管理型コンポーネントとの間の処理手順を例示するシーケンス図。

【図 3 9】

パネルコンポーネントにおけるベースコンポーネントとカスタムコンポーネントの関係を例示する図。

【図 4 0】

アプリケーションからパネルコンポーネントに対して決定操作がなされた時の処理を示すシーケンス図。

【図 4 1】

ユーザによるボタン操作に基づいてパネルコンポーネントの決定操作が行われた場合の処理を示すシーケンス図。

【図 4 2】

パネルコンポーネントと一覧検索型カスタムコンポーネントとの間の処理手順を示したシーケンス図。

【図 4 3】

テキストフィールドコンポーネントにおけるベースコンポーネントとカスタムコンポーネントの関係を例示したシーケンス図。

【図 4 4】

テキストフィールドコンポーネントにおけるベースコンポーネントとカスタムコンポーネント間の処理手順を示すシーケンス図。

【符号の説明】

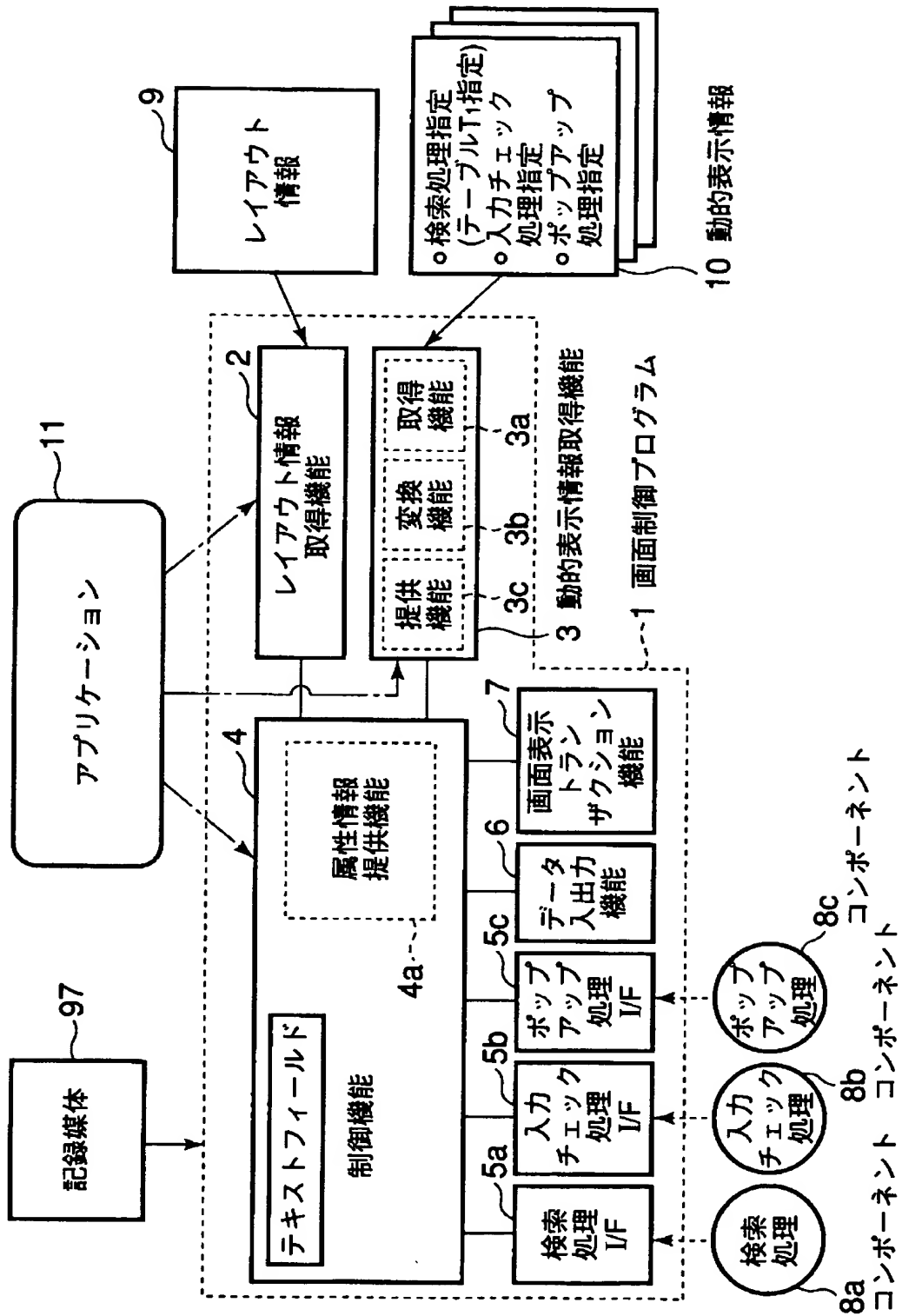
- 1、17、20、26、34、39…画面制御プログラム
- 2、18、24…レイアウト情報取得機能
- 3、25…動的表示情報取得機能
- 3 a…取得機能
- 3 b…変換機能

- 3 c …提供機能
- 4、4 0 …制御機能
- 4 a …属性情報提供機能
- 5 a、2 7 …検索処理インタフェース
- 5 b、3 1 …入力チェック処理インタフェース
- 5 c …ポップアップ処理インタフェース
- 6 …データ入出力機能
- 7、3 5 …画面表示トランザクション機能
- 8 a ～ 8 c …コンポーネント
- 9、1 9 …レイアウト情報
- 1 0 …動的表示情報
- 1 1 …アプリケーション
- 2 1 …計算機
- 2 2、2 9、3 7 …ネットワーク
- 2 3 …WWWサーバ
- 2 8 …検索処理スタブ
- 3 0 …検索処理リモートオブジェクト
- 3 2 …入力チェック処理スタブ
- 3 3 …入力チェック処理リモートオブジェクト
- 3 6 …画面表示トランザクション情報送受信部
- 3 8 …画面表示トランザクション管理サーバ
- 4 1 …業務処理インタフェース
- 4 2 …業務処理用のコンポーネント
- 5 9、6 0 …インタフェース機能

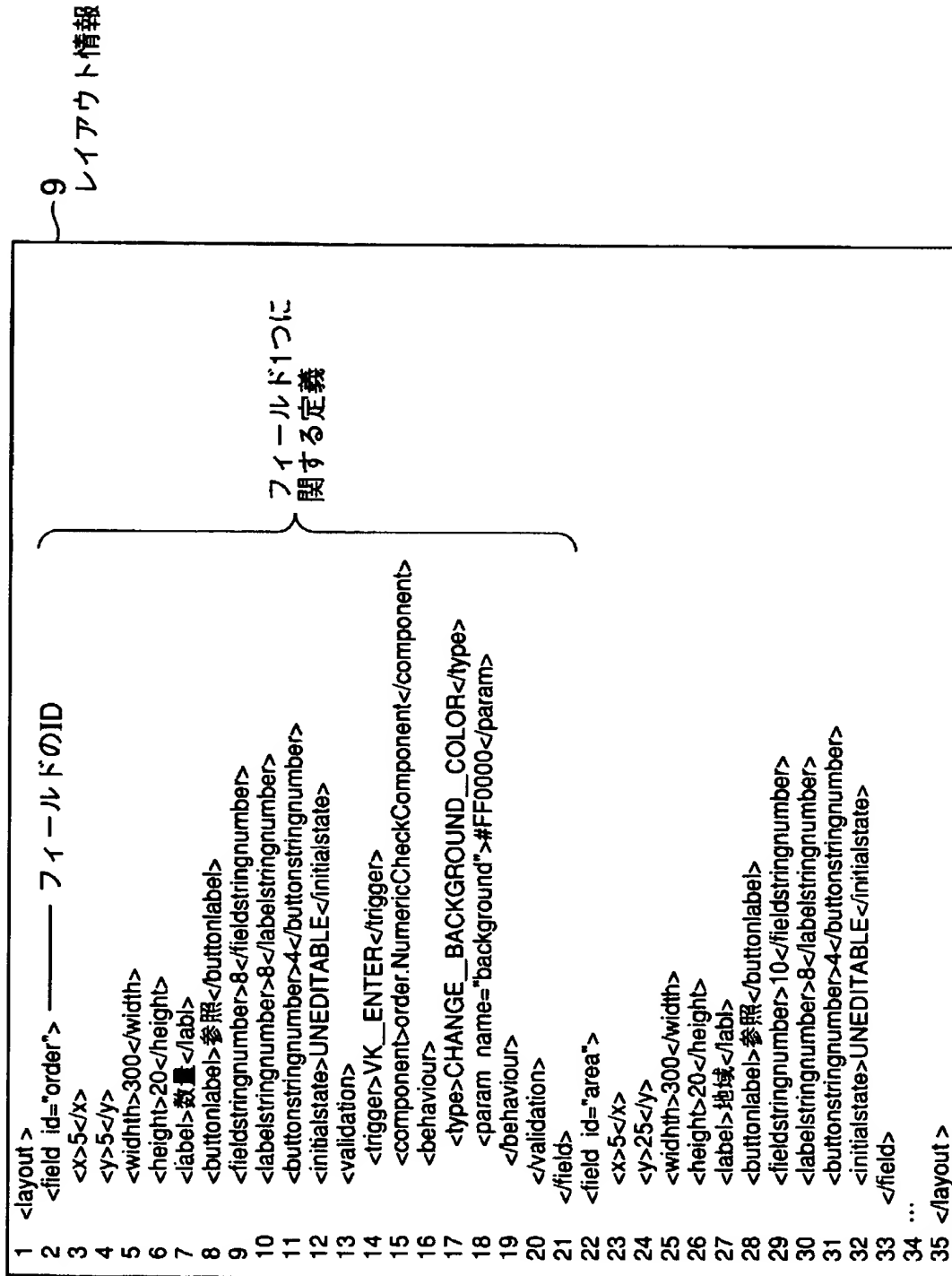
【書類名】

図面

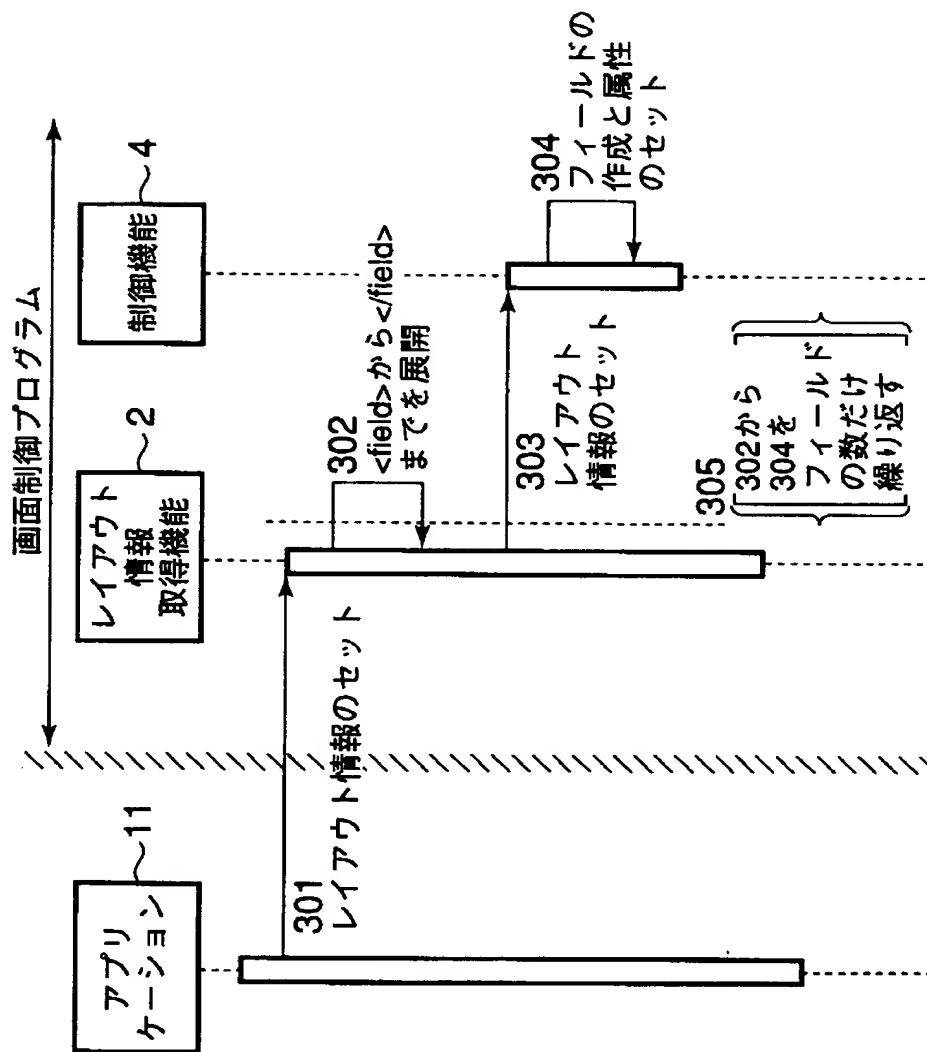
【図 1】



【図 2】

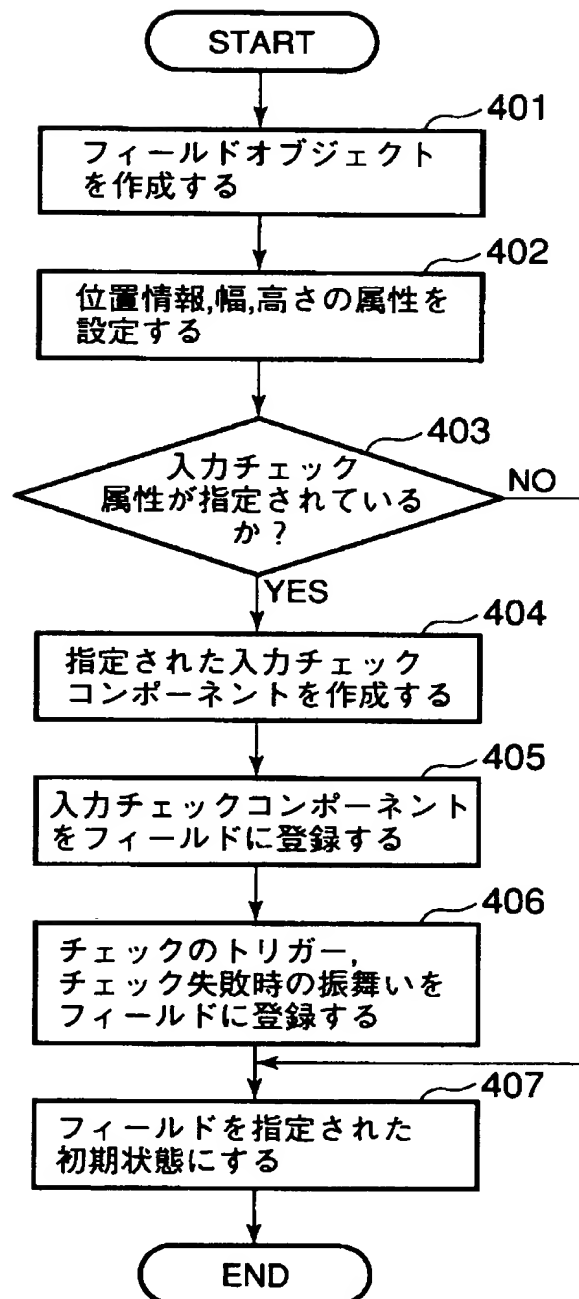


【図 3】





【図 4】



【図 5】

The diagram illustrates a software interface window labeled "12 画面". The window has a standard title bar with three control buttons (minimize, maximize, close) in the top right corner. The main content area contains three vertically stacked dashed rectangular input fields. To the left of these fields are labels: "数量" (Quantity) for the top field, "地域" (Region) for the middle field, and "商品" (Product) for the bottom field. These labels are grouped by a bracket and collectively labeled "12a", "12b", and "12c" respectively. To the right of each dashed field are labels "12d", "12e", and "12f" respectively, connected by dashed lines. At the bottom right of the window, there are three rectangular buttons labeled "OK", "キャンセル" (Cancel), and "終了" (End).

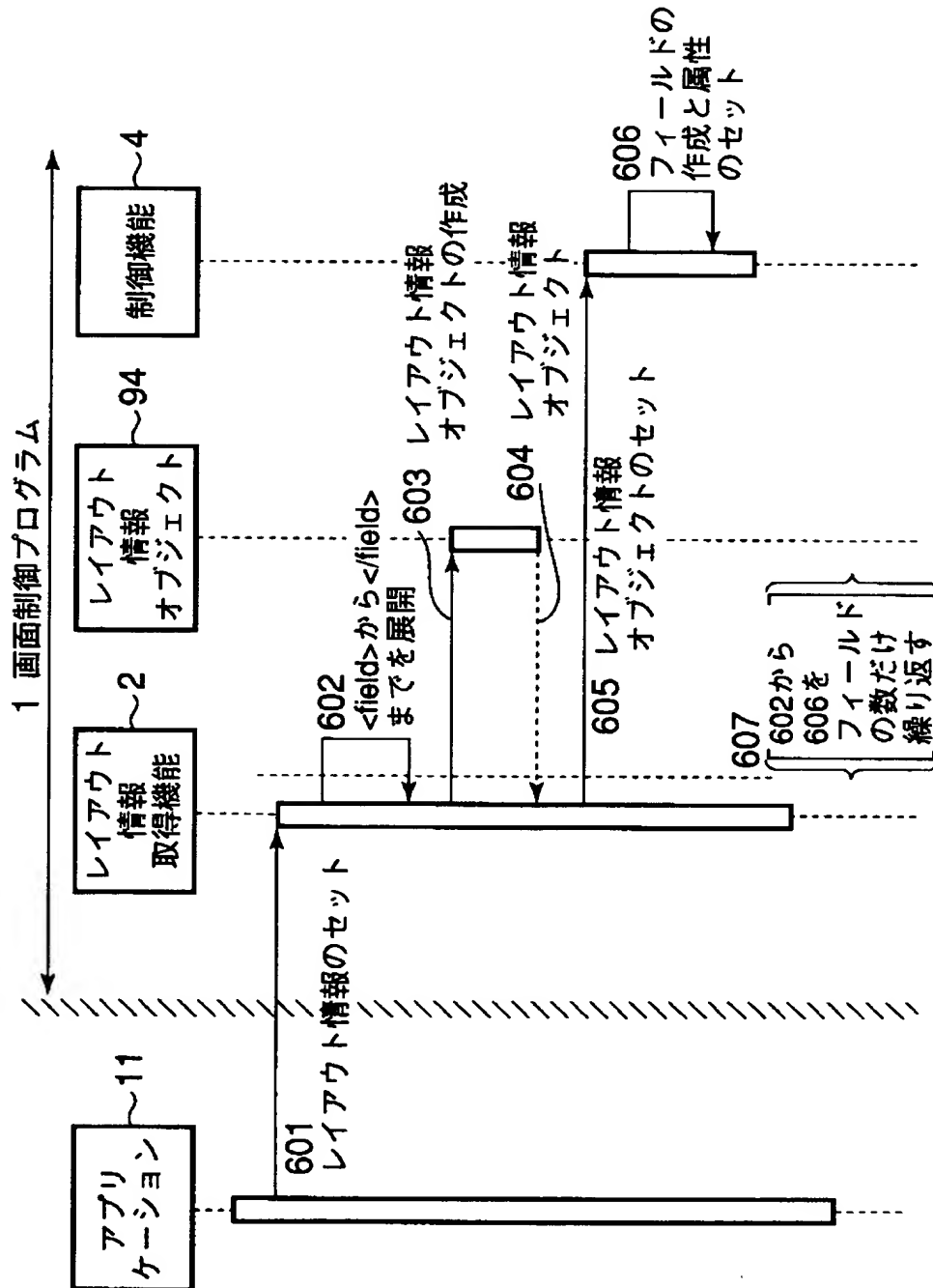
数量  
12a  
地域  
12b  
商品  
12c

12d  
12e  
12f

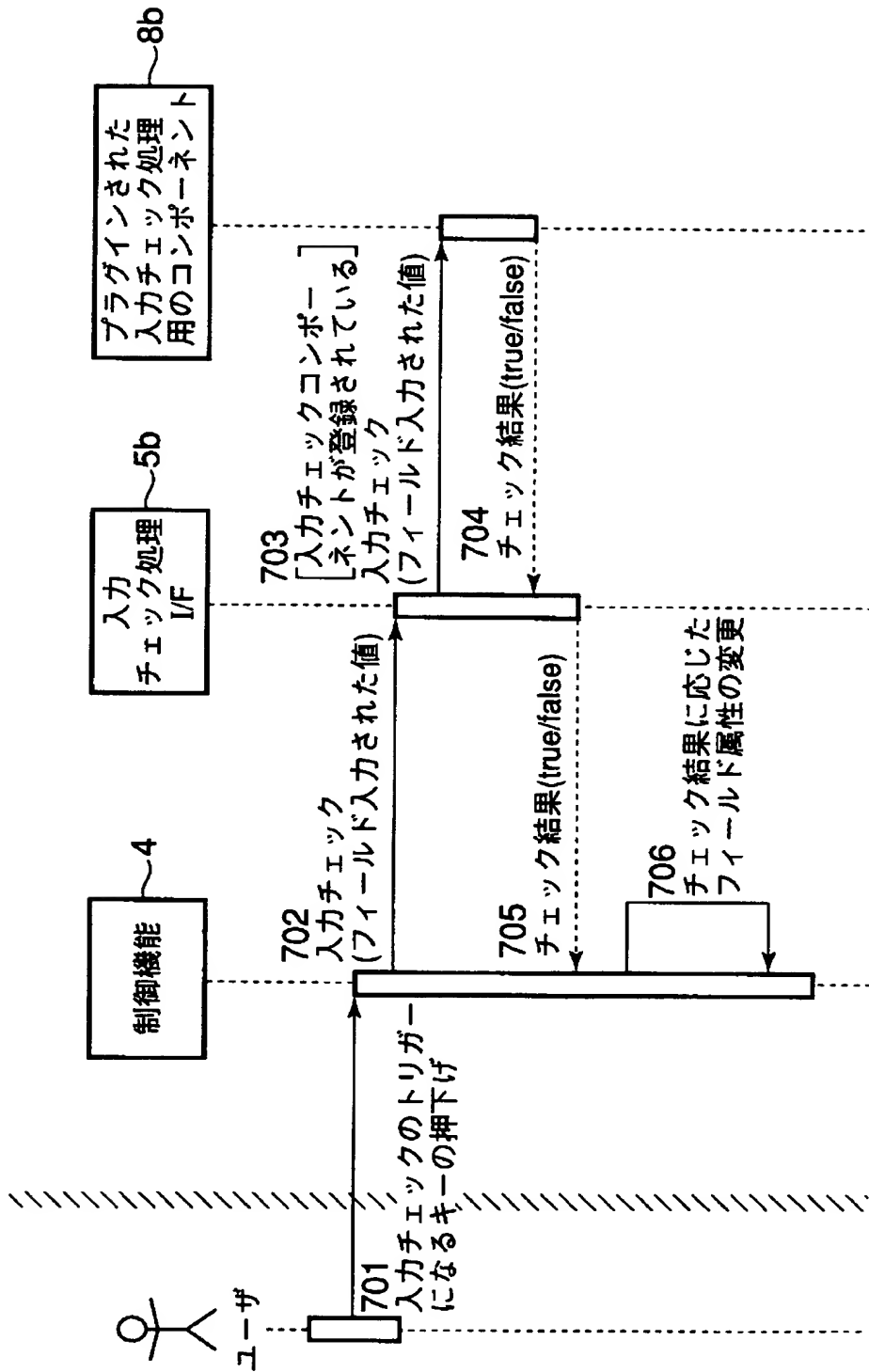
OK キャンセル 終了

12 画面

【図 6】



【図 7】



【図 8】

数量 A 12d

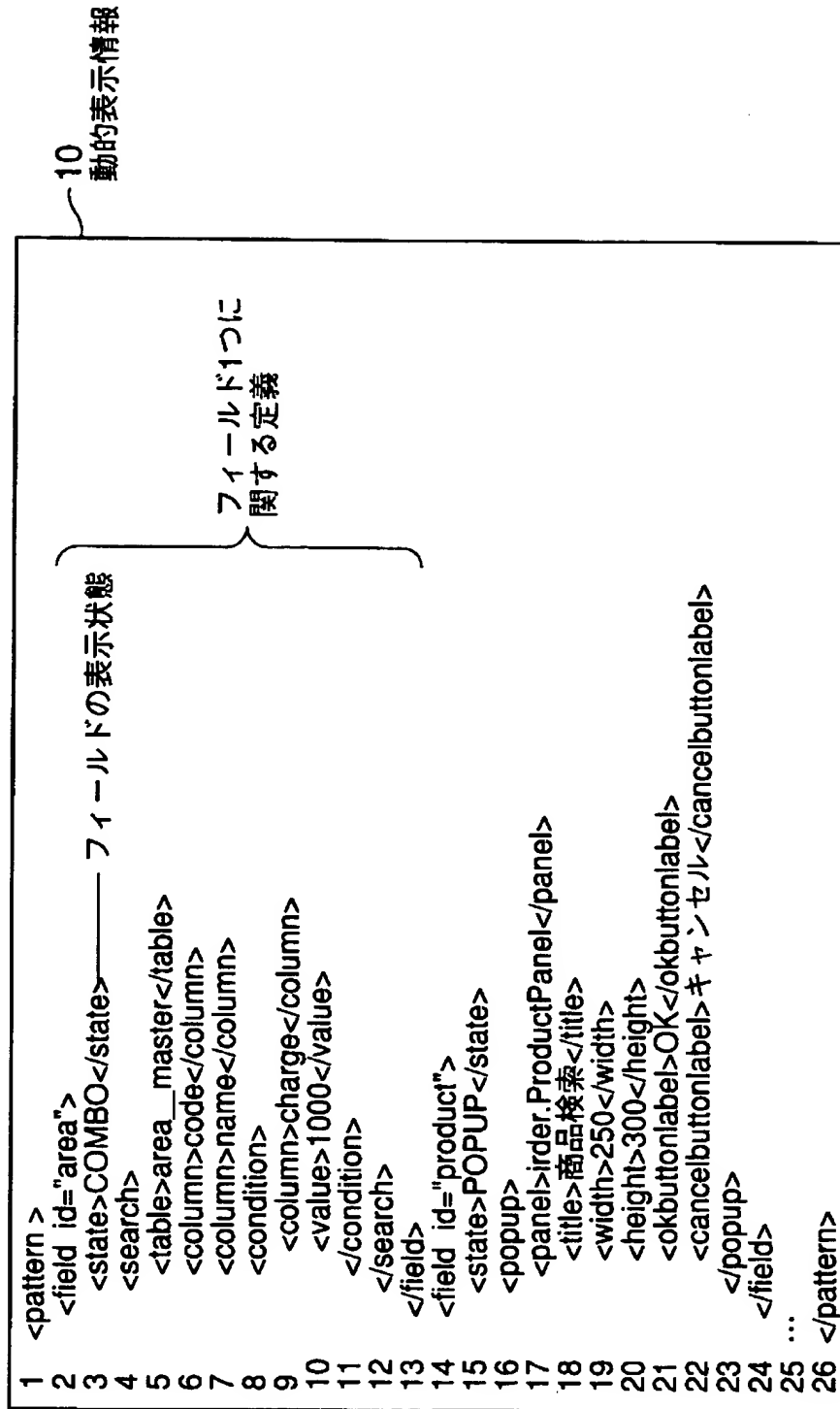
地域  ▼

商品  参照

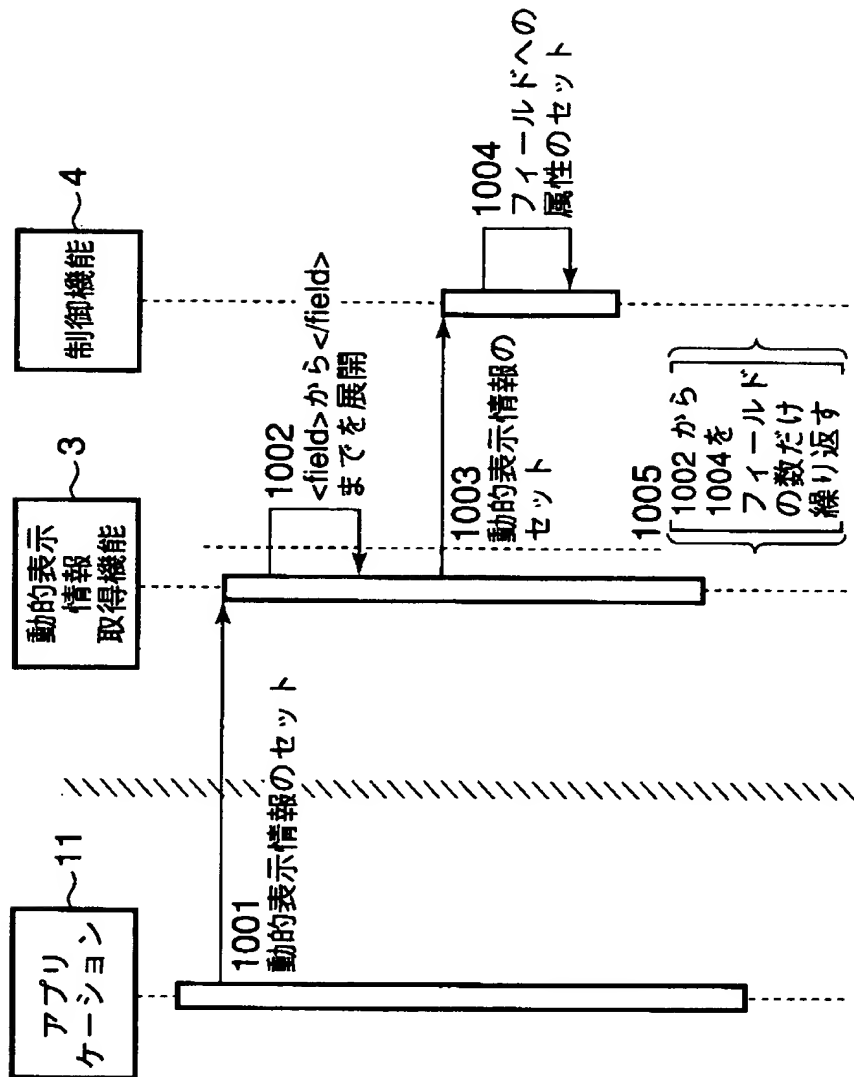
OK キャンセル 終了

12 画面

【図 9】



【図 1 0】



【図 1 1】

数量  12d

地域  ▼ 14a

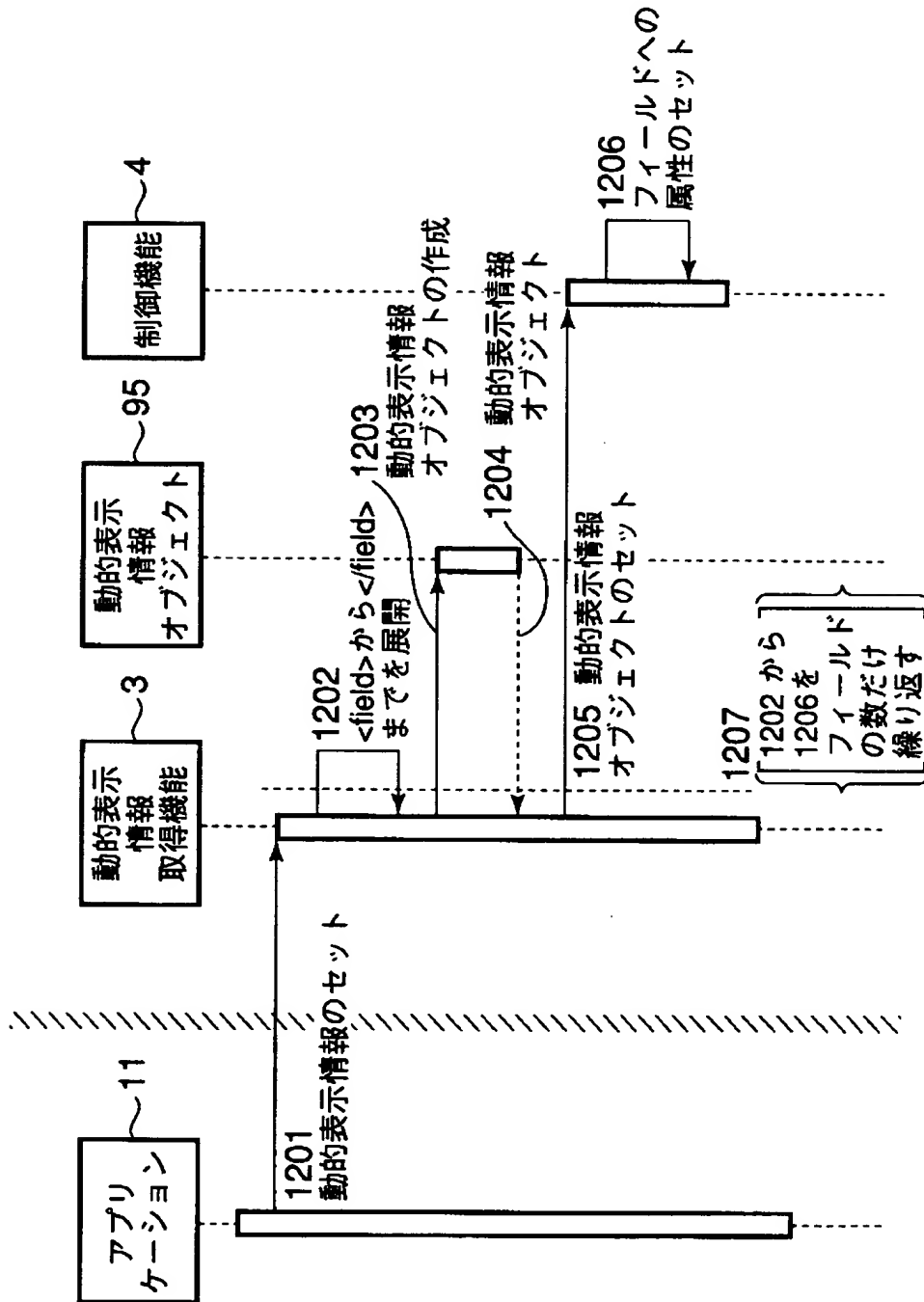
商品  12f 参照 14b

12 画面

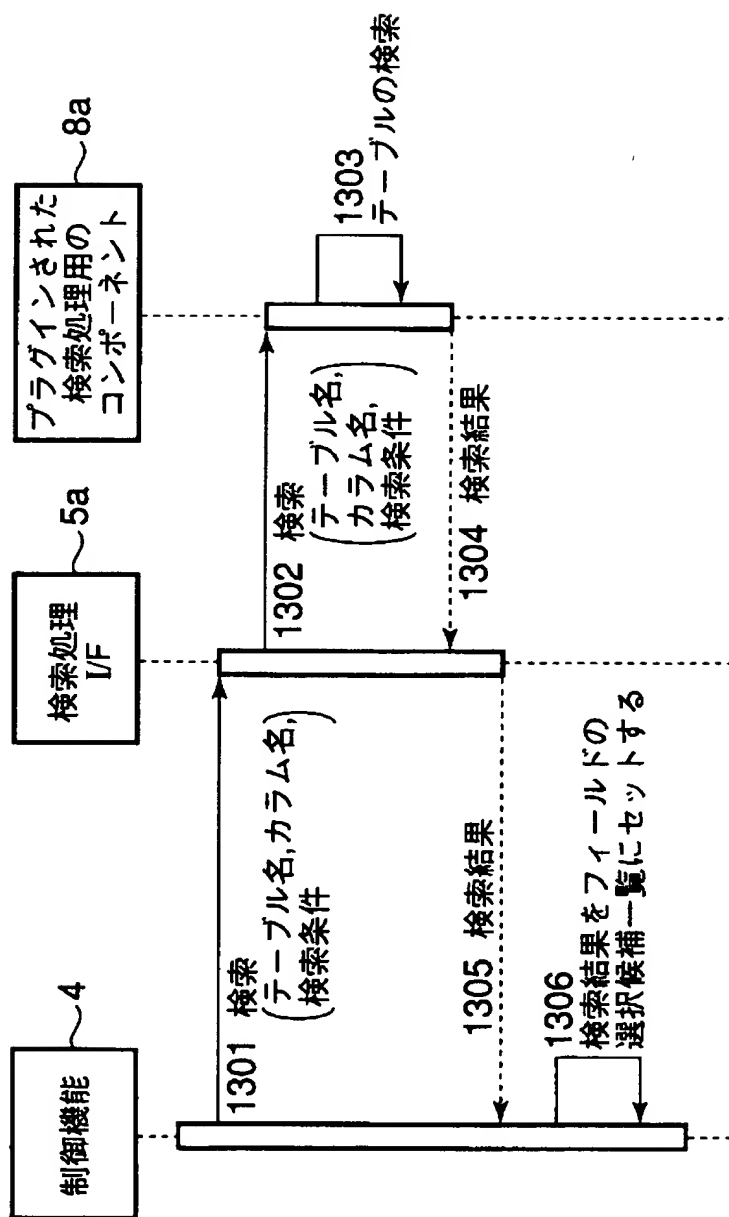
OK キャンセル 終了



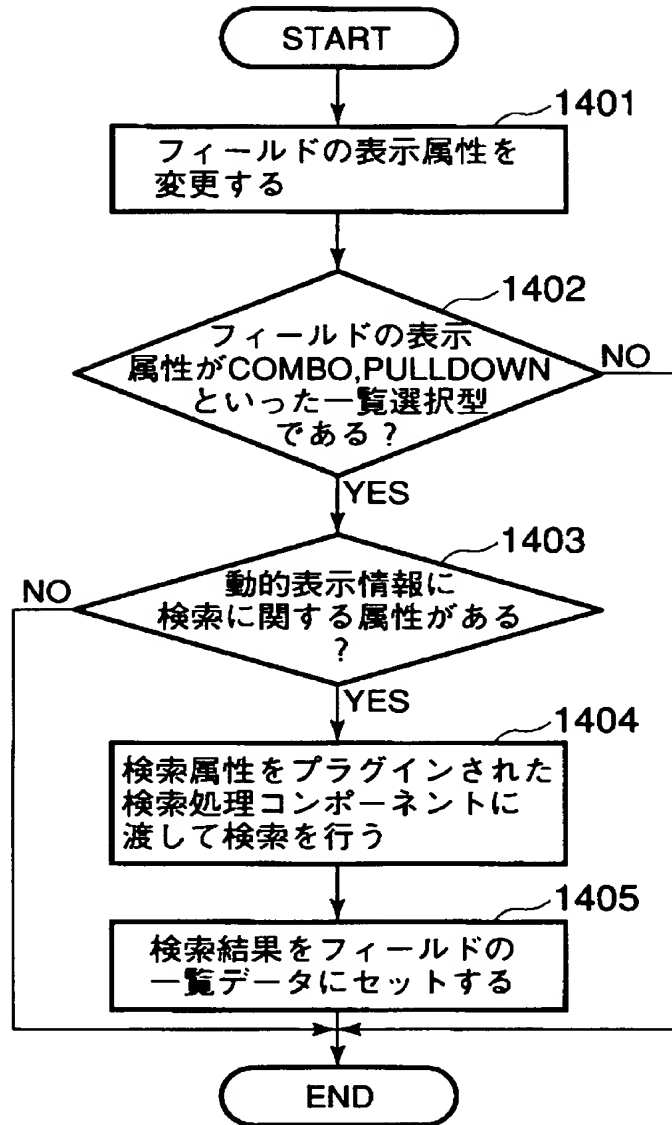
【図 12】



【図 1 3】



【図 1.4】



【図 1 5】

数量

地域

商品

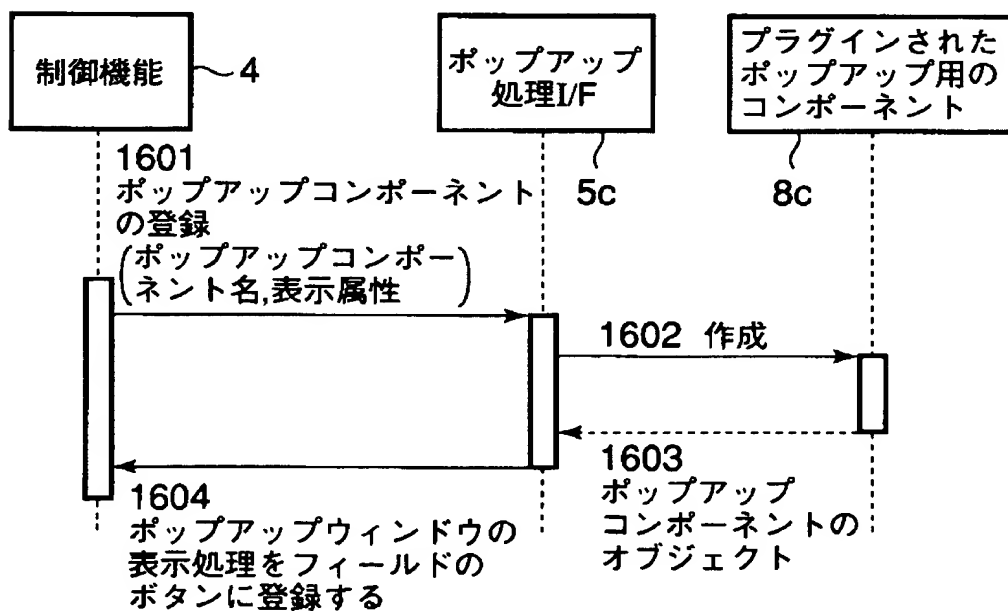
001 北海道  
002 青森  
003 岩手  
004 秋田

参照

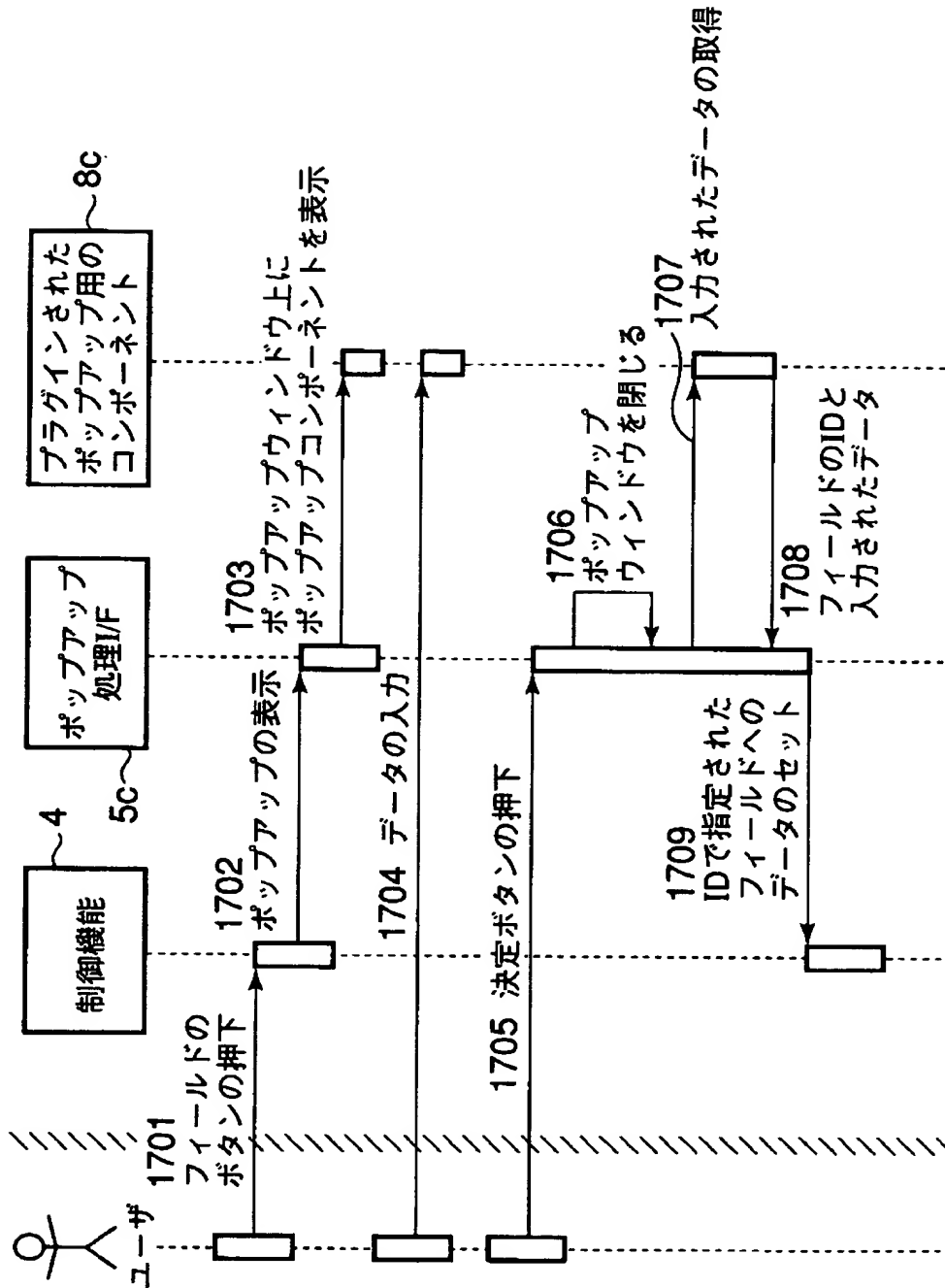
OK キャンセル 終了

12 画面

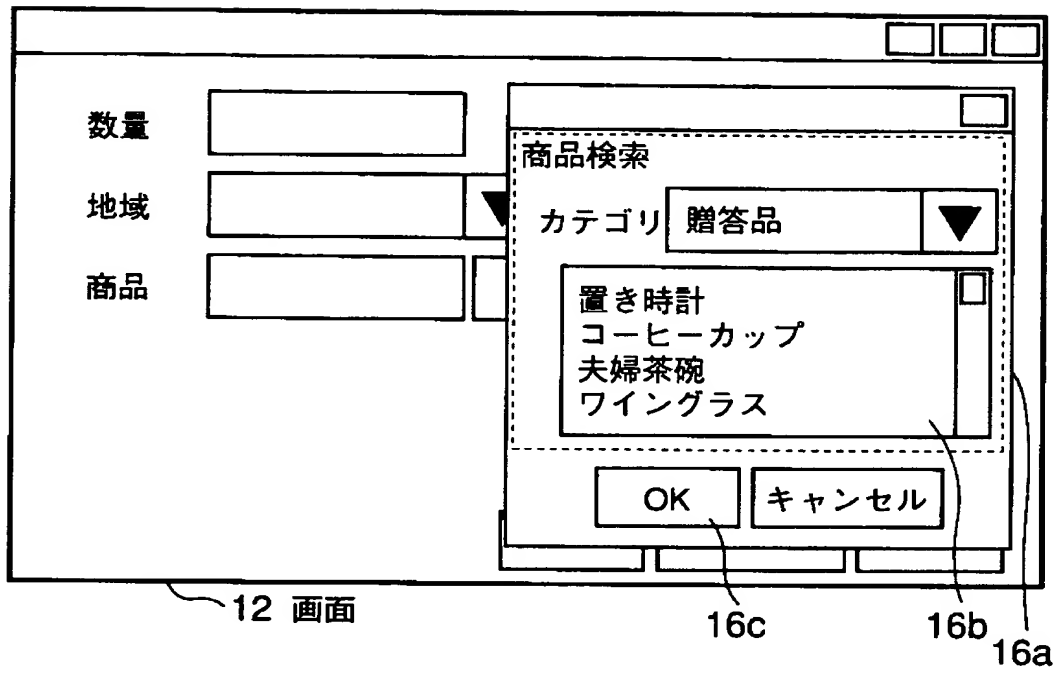
【図 1 6】



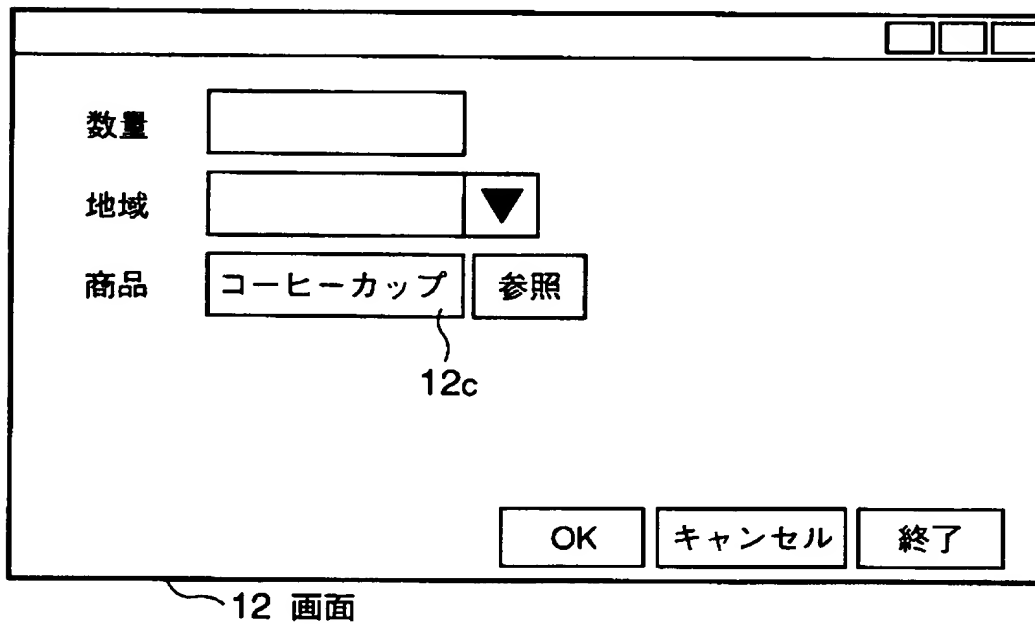
【図 1 7】



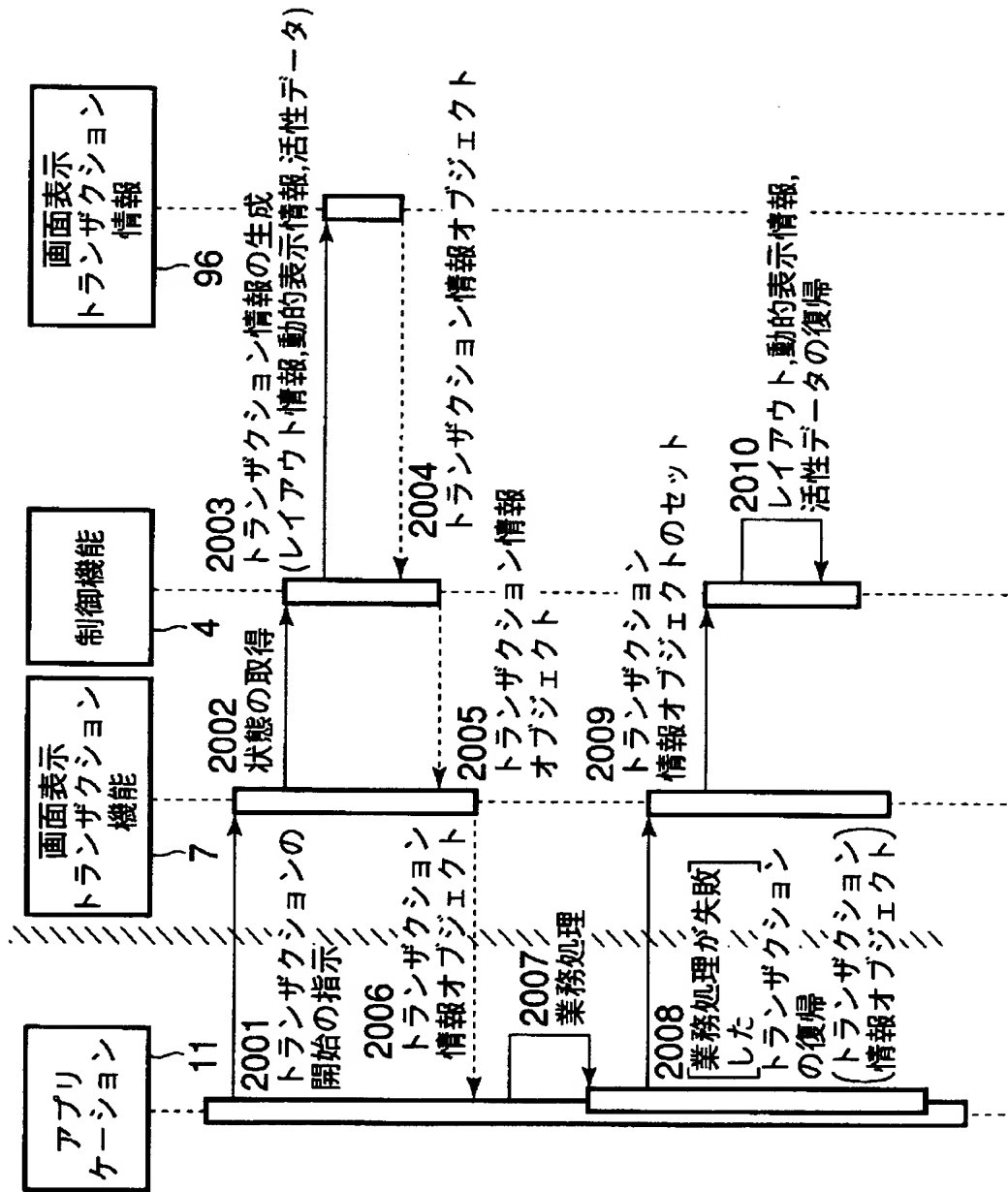
【図 1 8】



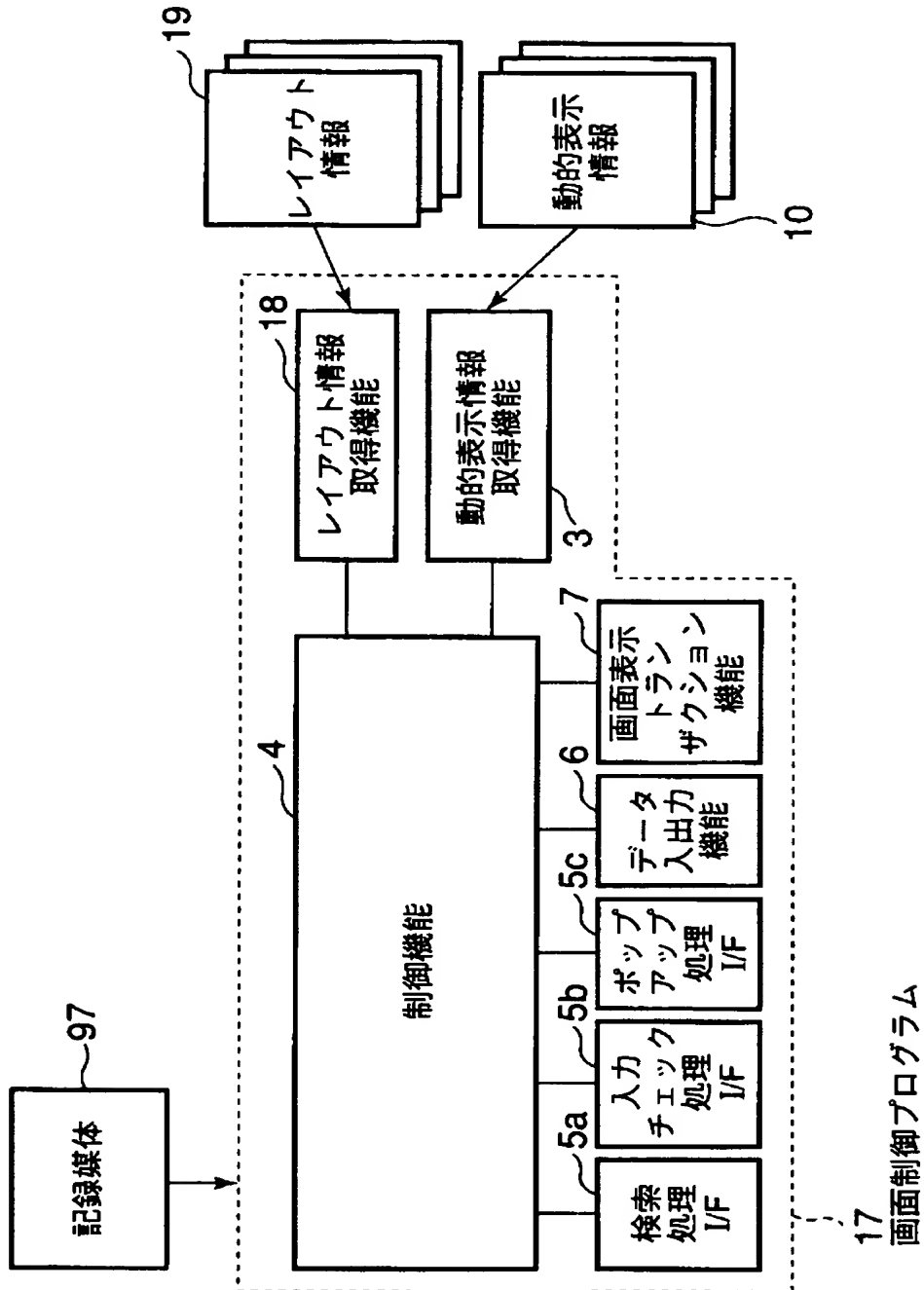
【図 1 9】



【図 20】

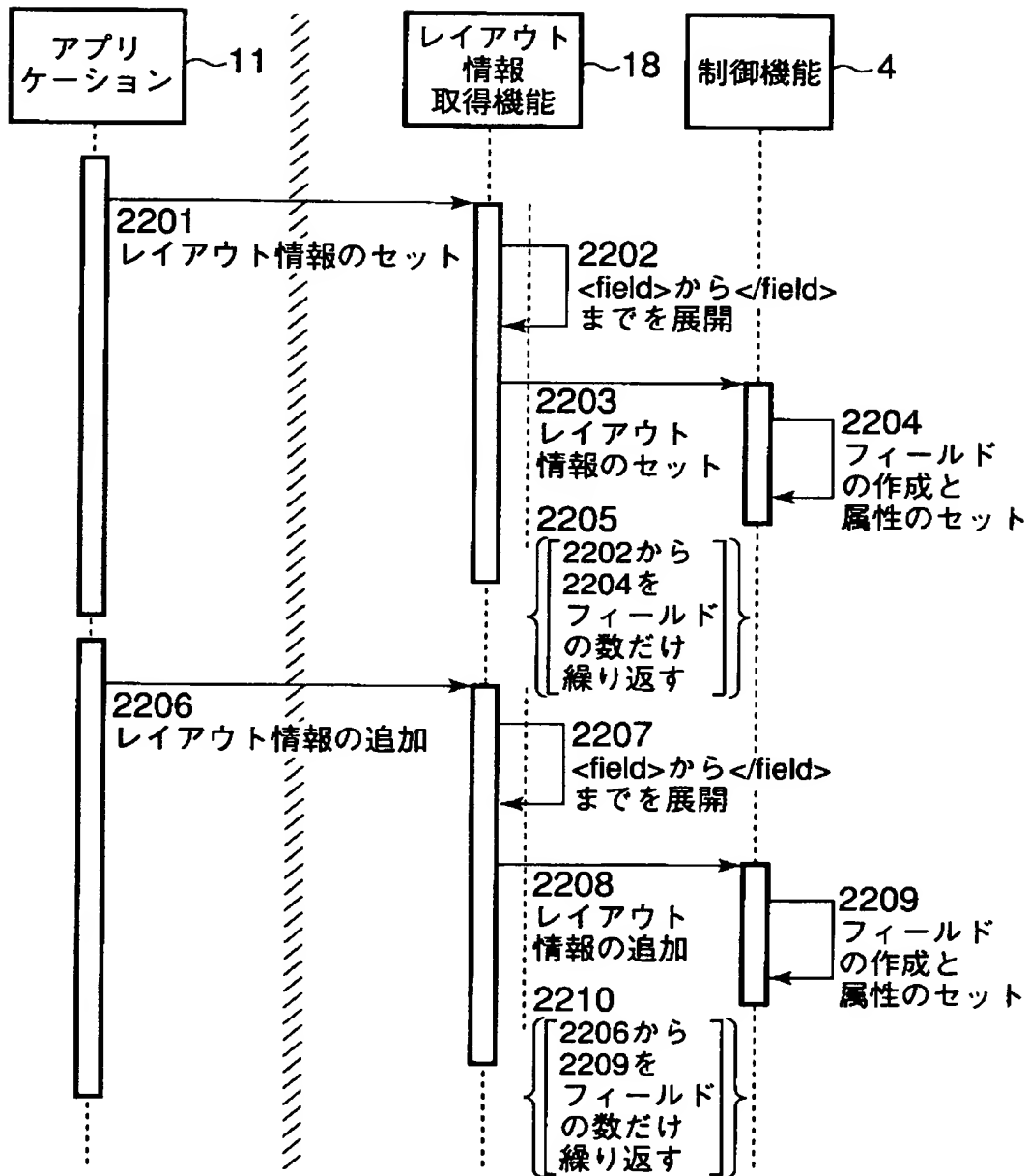


【図 2 1】

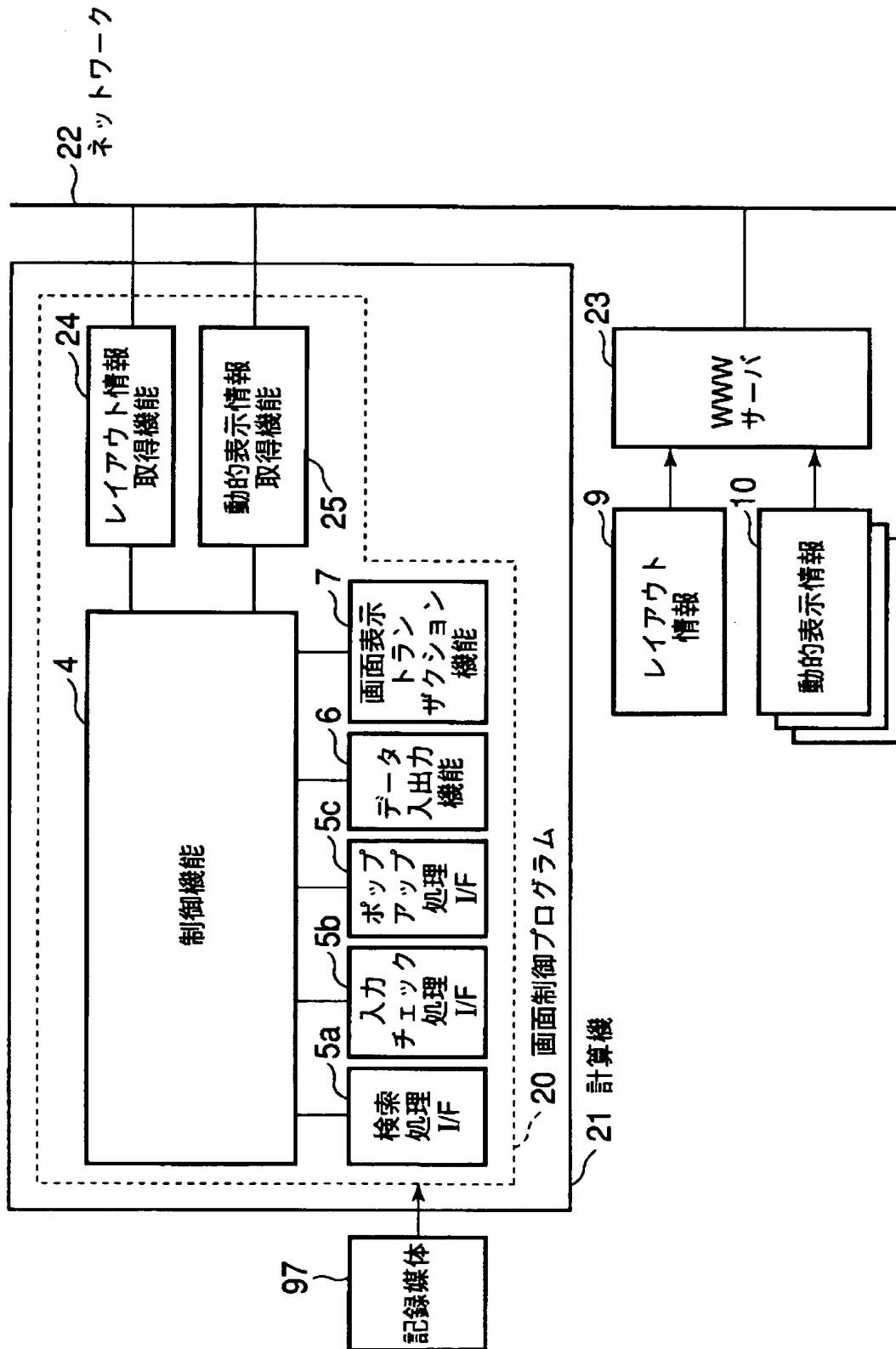




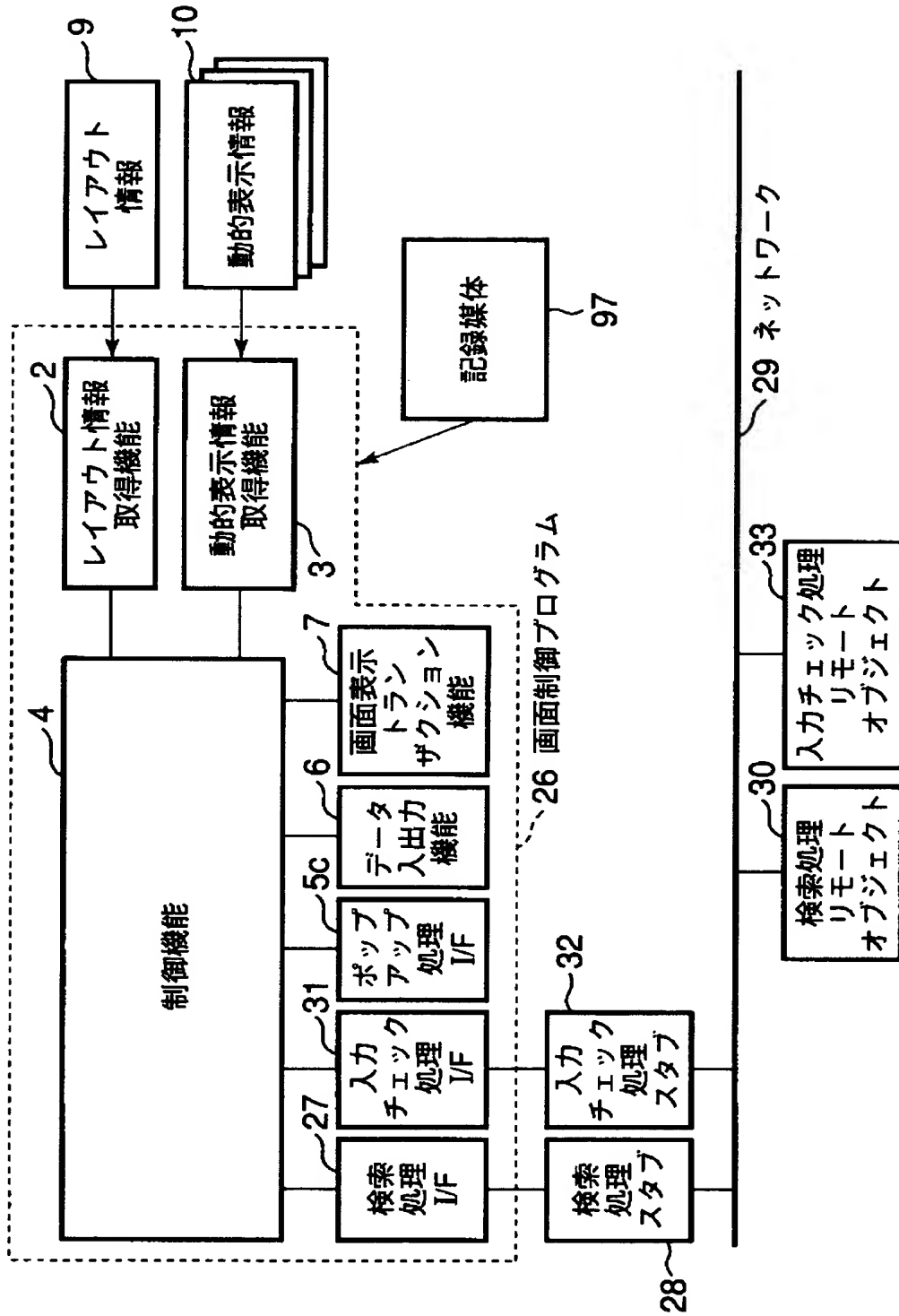
【図 2 2】



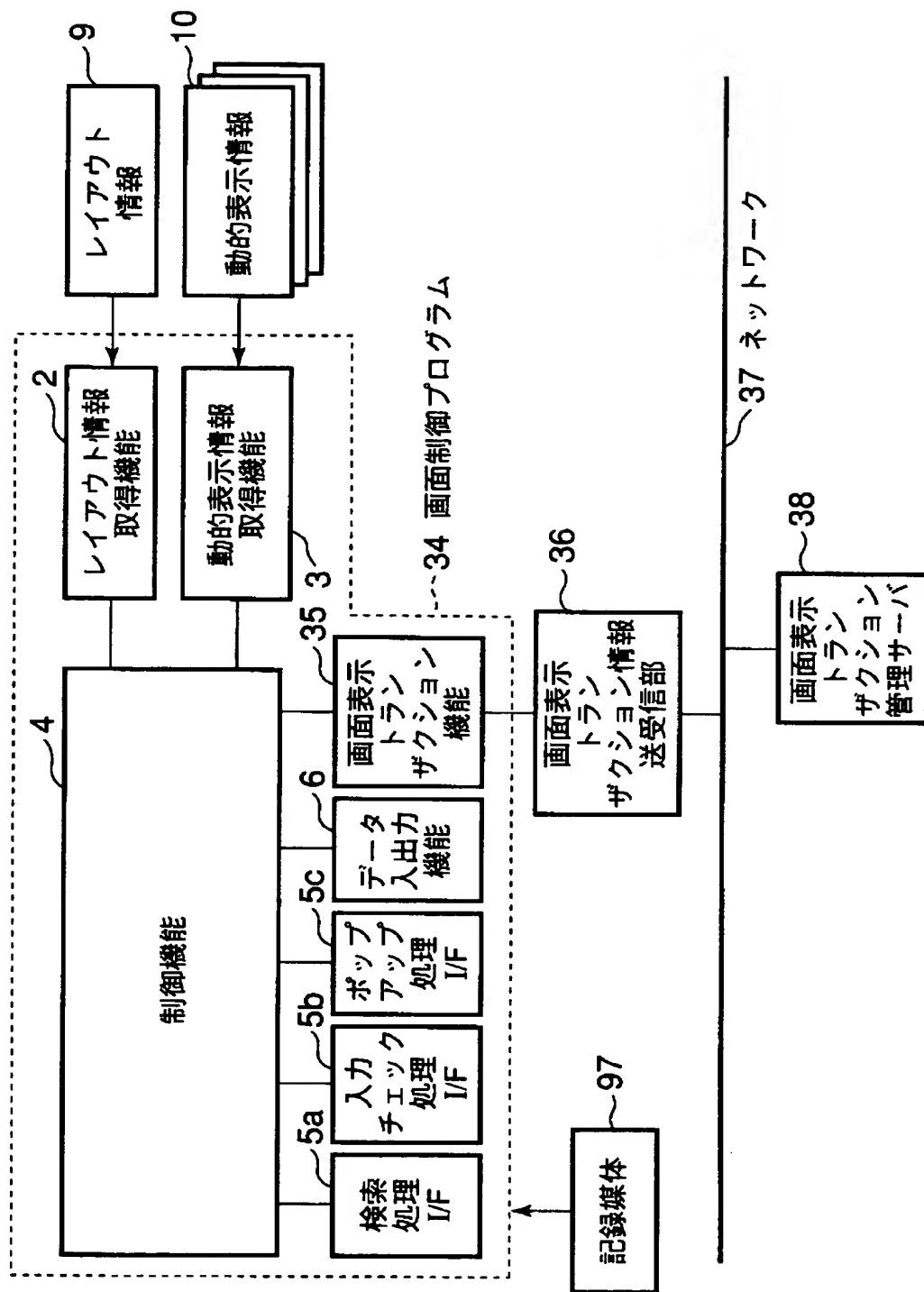
【図 23】



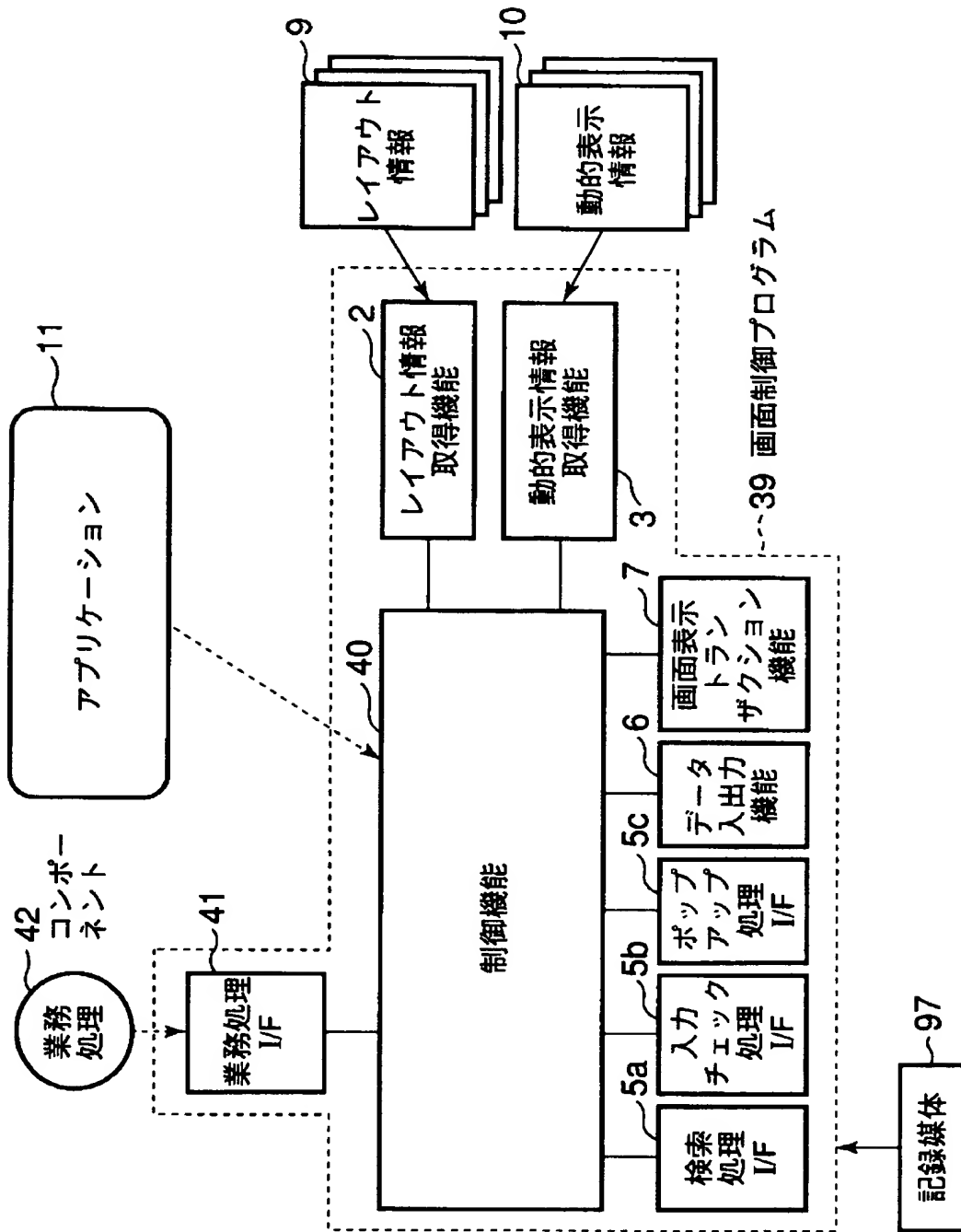
【図 24】



【図 2 5】



【図 26】



【図 2 7】

税込み金額の検索条件

国  44

振り込み先

設備費別  45

税込金額  48

取引相手  46

検索

画面パターン 値のセット データ取得 APIでパターン

43 画面

【図 2 8】

税込み金額の検索条件

国  44

振り込み先  45

設備費別

税込金額  48

計上年月  49 年  50

取引相手  46

検索

画面パターン 値のセット データ取得 APIでパターン

47 画面

【図 2 9】

税込み金額の検索条件

国

振り込み先

設備費別

税込金額

計上年月

取引先相手先

0000 JAPAN

0001 USA

0002 CANADA

0003 KOREA

0004 CHINA

0005 INDIA

0006 AUSTRALIA

0007 SPAIN

画面パターン

値のセット

データ取得

APIでパターン

51 画面

【図 3 0】

税込み金額の検索条件 SS

国

振り込み先

設備費別

税込金額

計上年月

取引先相手先

0000 BANK A

0001 BANK B

0002 BANK C

0003 BANK D

0004 BANK E

0005 BANK F

0006 BANK G

0007 BANK H

画面パターン

値のセット

データ取得

53 画面

【図 3 1】

税込み金額の検索条件 SS

国 45 0001 USA ▼

振り込み先 0003 BANK D 検索

設備費別 ▼

税込金額

計上年月 ▼ 年 ▼

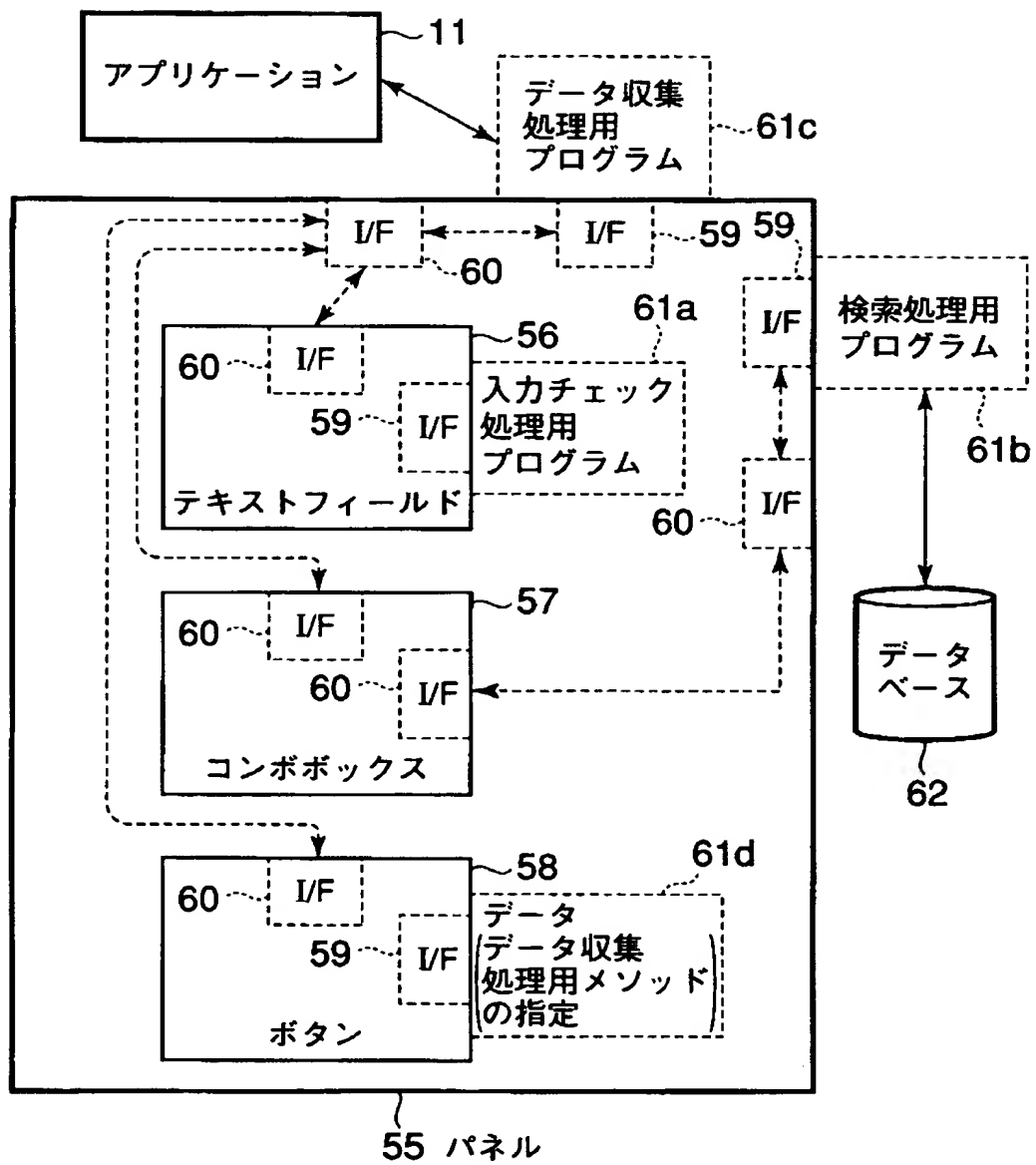
取引先相手先 ▼

画面パターン 値のセット データ取得 APIでパターン

53



【図 3 2】



【図 3 3】

**フレーム(E) 編集(E)** 検索(S) 表示(V) プレビュー(P) フォント(F) ウィンドウ(W) ヘルプ(H)

Swing Containers Data Express dbSwing More dbSwing dbSwing Models JBCL LBCJ Containers AWT KL Group Other

実行ファイル対象 : Application1.java

プロジェクト

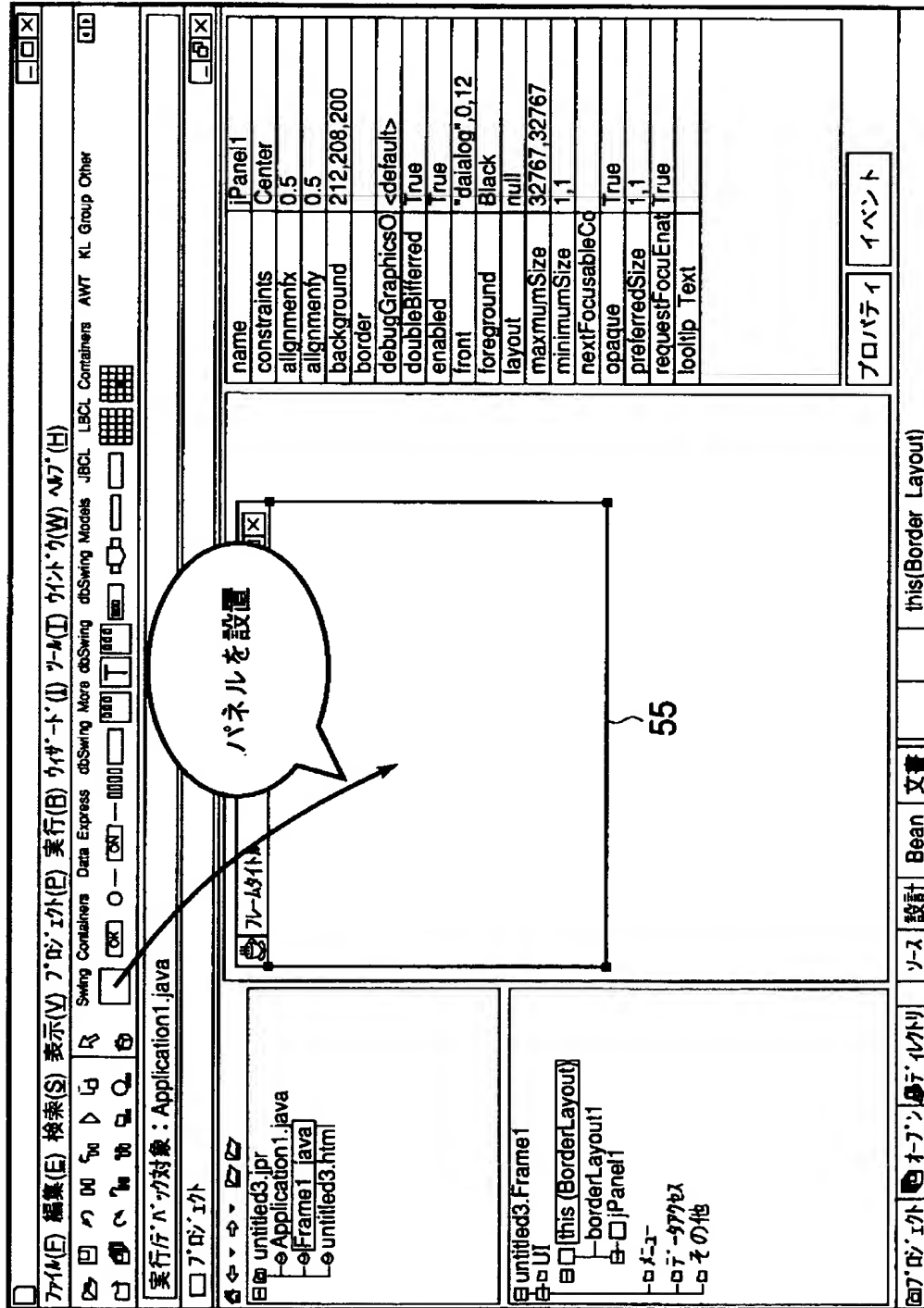
```

Application1.java
├── Frame1.java
└── untitled3.html

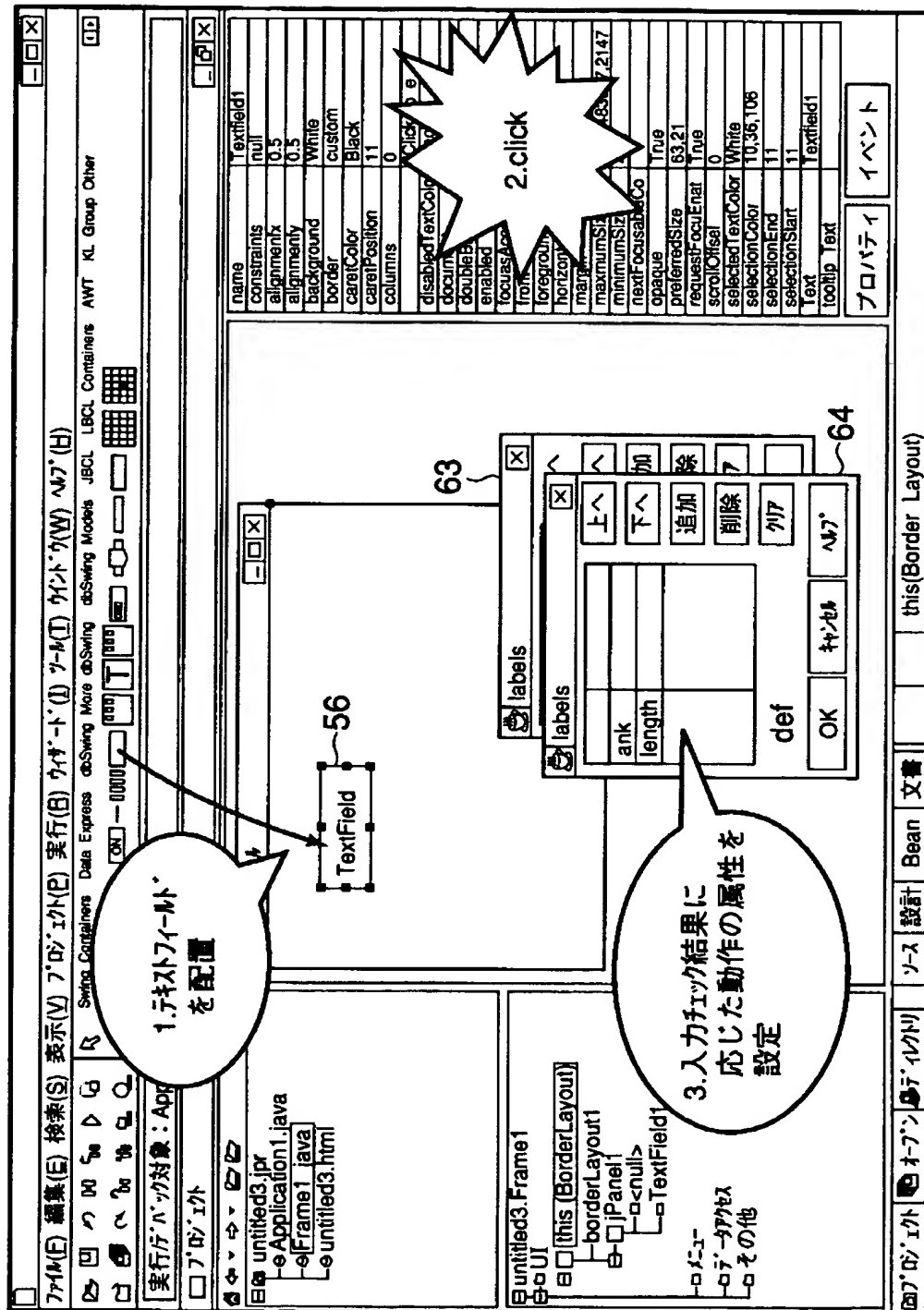
Application1.java
public class Application1 extends JFrame {
    public Application1() {
        setTitle("Application1");
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());
        // ... other code ...
    }
}

```

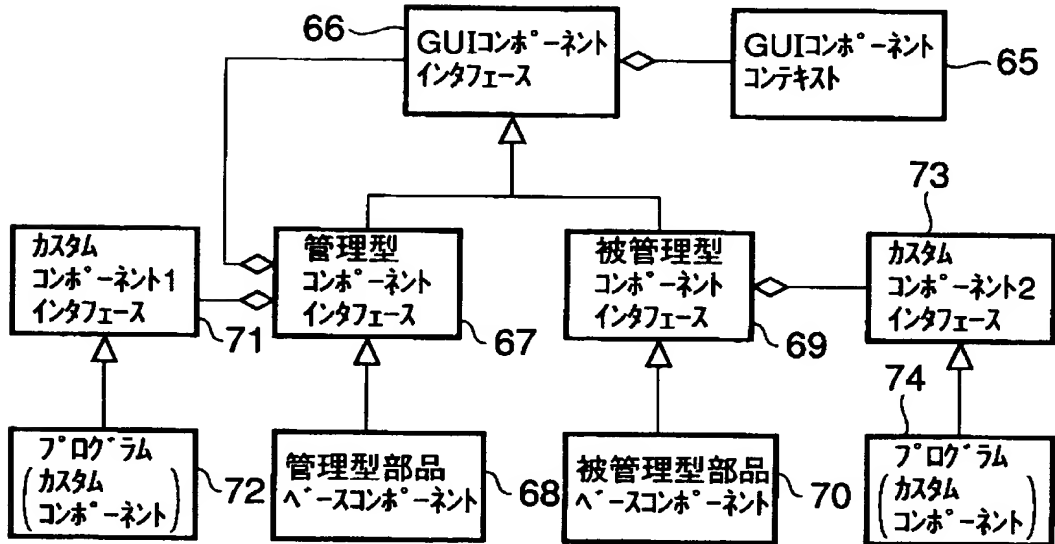
【図 34】



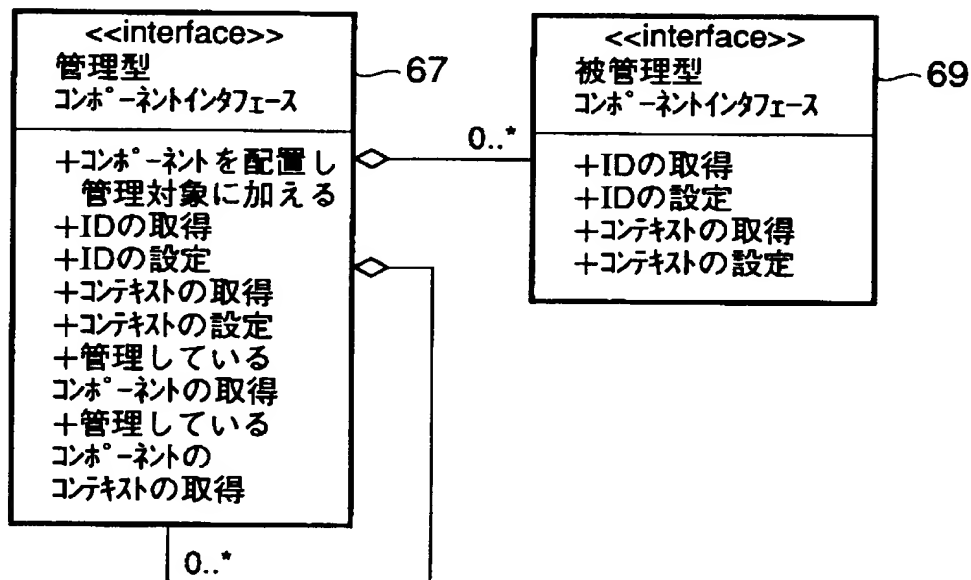
【図 3 5】



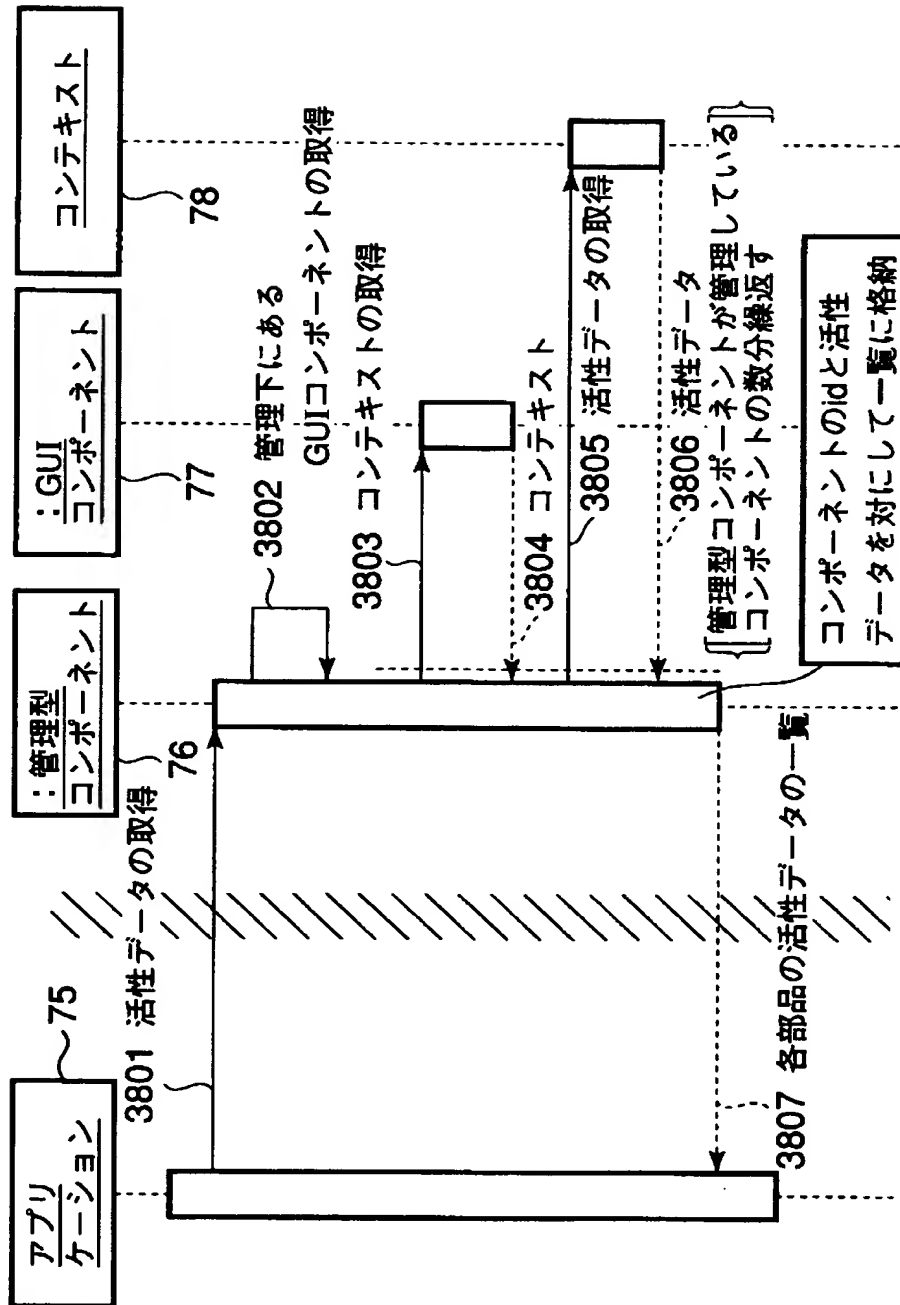
【図 3 6】



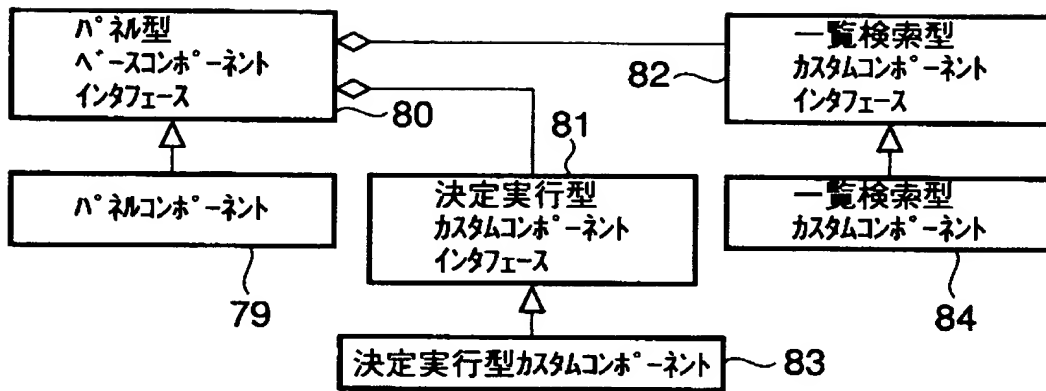
【図 3 7】



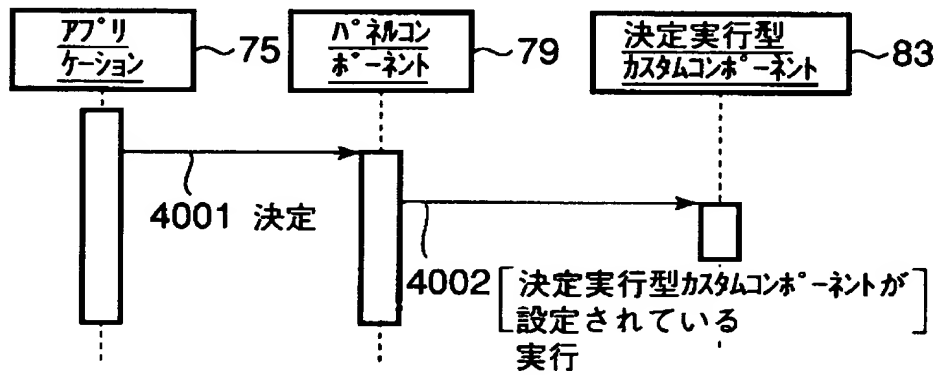
【図 3 8】



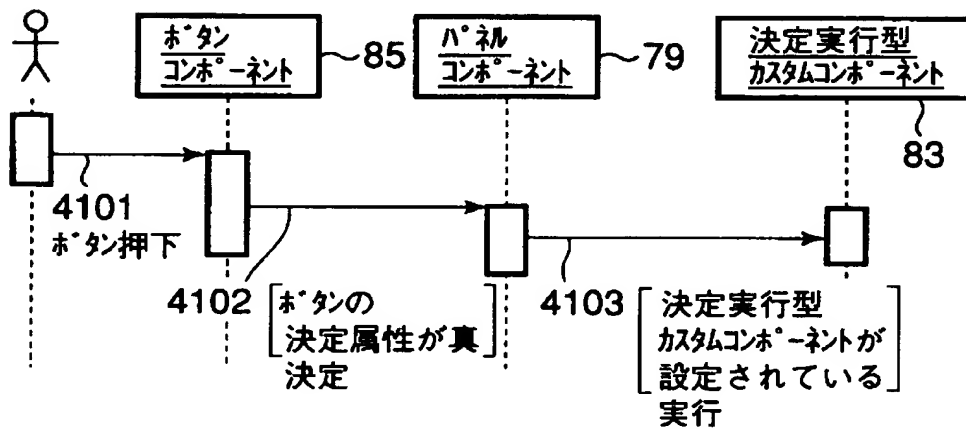
【図 3 9】



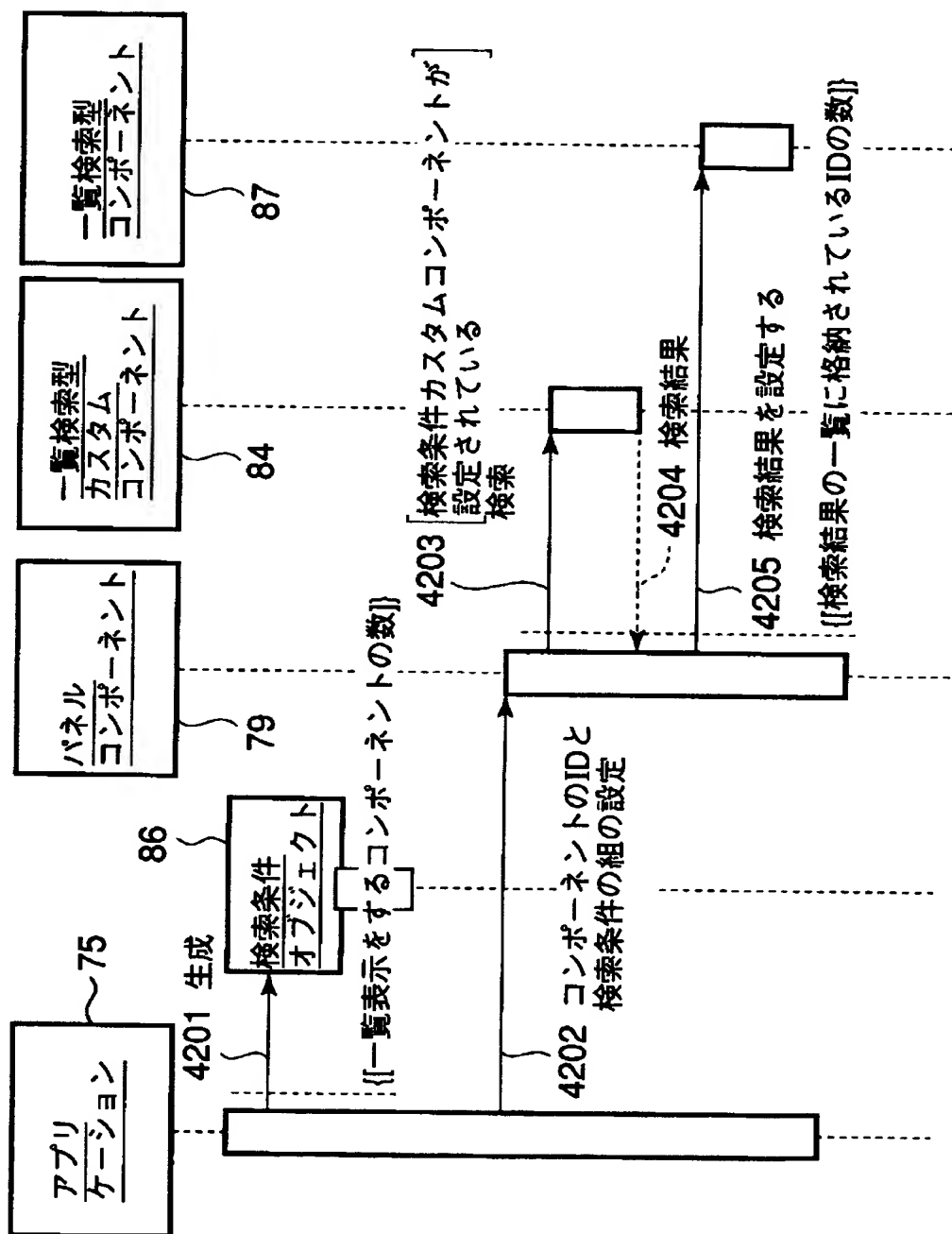
【図 4 0】



【図 4 1】

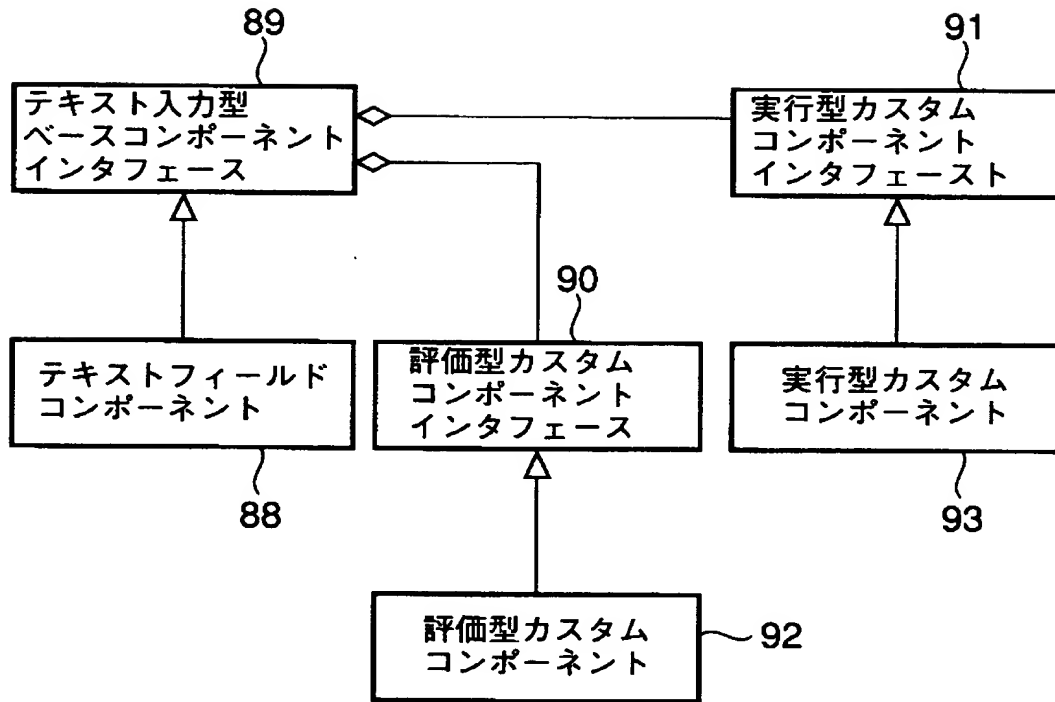


【図 4 2】

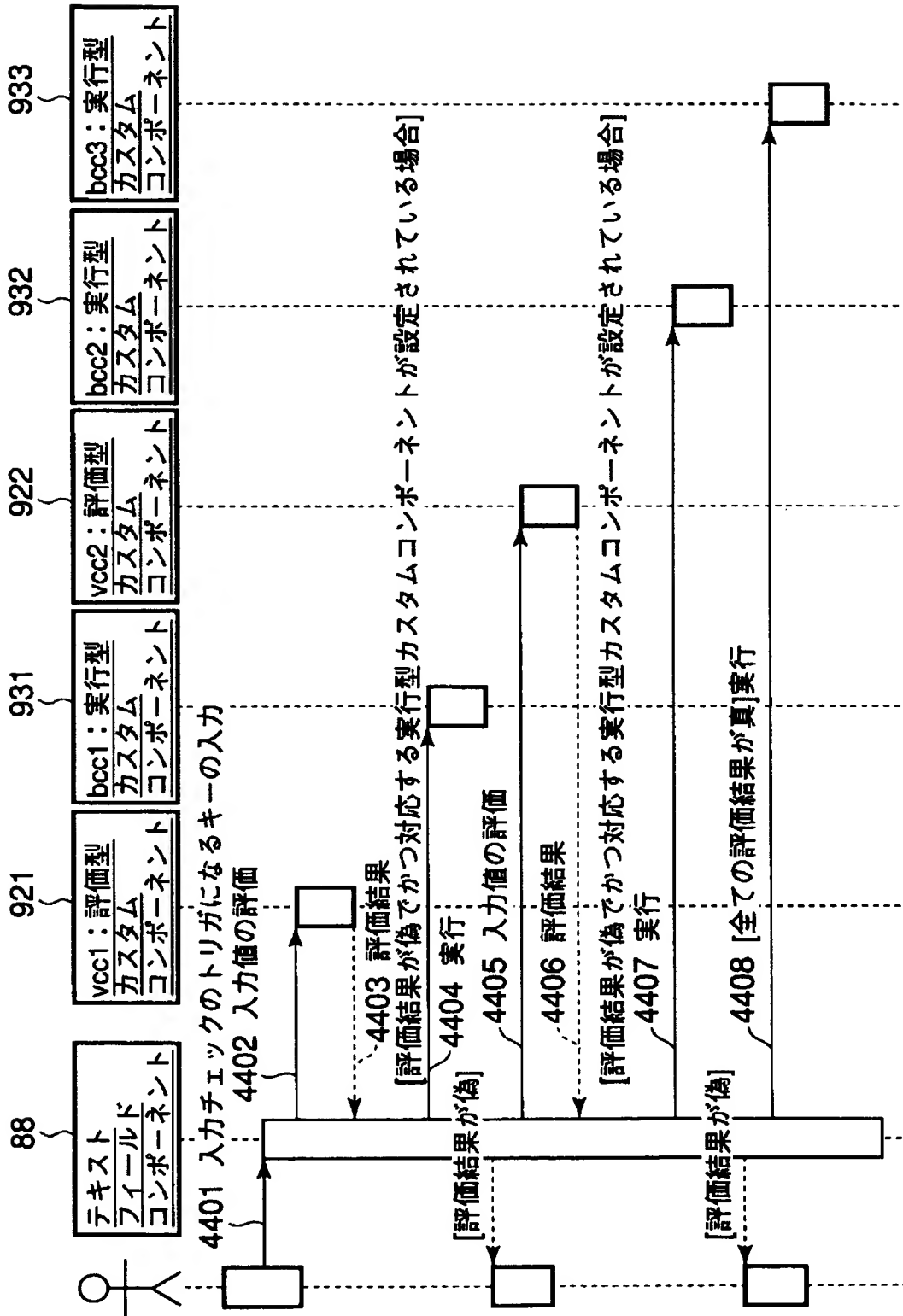




【図 4 3】



【図 4 4】



【書類名】 要約書

【要約】

【課題】 画面プログラムの開発・変更を効率化する。

【解決手段】 コンピュータに、動的に変化する表示内容を得るために予め用意されている動的表示処理を指定する動的表示処理識別情報と、この動的表示処理識別情報で指定されている動的表示処理で利用する表示属性情報とを取得する動的表示情報取得機能 3、動的表示情報取得機能 3 によって取得された動的表示処理識別情報で指定されている動的表示処理に対してその動的表示処理に対応する表示属性情報を提供する属性情報提供機能 4 a を実現させるための画面制御プログラム 1。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [ 0 0 0 0 0 3 0 7 8 ]

1. 変更年月日 1 9 9 0 年 8 月 2 2 日

[変更理由] 新規登録

住 所 神奈川県川崎市幸区堀川町 7 2 番地

氏 名 株式会社東芝